UNIVERSIDADE FEDERAL DE UBERLÂNDIA
INSTITUTO DE LETRAS E LINGUÍSTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTUDOS LINGUÍSTICOS

HEITOR CARVALHO DE ALMEIDA NETO

GEConWeb*: desenvolvimento de uma plataforma *on-line* para exploração de *corpora

UBERLÂNDIA
2023

HEITOR CARVALHO DE ALMEIDA NETO

***GEConWeb*:** desenvolvimento de uma plataforma *on-line* para exploração de *corpora*

Dissertação apresentada ao Programa de Pós-Graduação em Estudos Linguísticos, no curso de Mestrado em Estudos Linguísticos, do Instituto de Letras e Linguística, da Universidade Federal de Uberlândia, como requisito para obtenção do título de Mestre em Estudos Linguísticos.

Área: Estudos em Linguística e Linguística Aplicada.

Linha de Pesquisa (1) Teoria, descrição e análise linguística.

Orientador: Prof. Dr. Ariel Novodvorski.

UBERLÂNDIA
2023

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

A447g
2023

Almeida Neto, Heitor Carvalho de, 1984-
GEConWeb [recurso eletrônico] : desenvolvimento de uma plataforma *on-line* para exploração de *corpora* / Heitor Carvalho de Almeida Neto. - 2023.

Orientador: Novodvorski Ariel.
Dissertação (mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Estudos Linguísticos.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2023.7034
Inclui bibliografia.

1. Linguística. I. Ariel, Novodvorski, 1968-, (Orient.). II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Estudos Linguísticos. III. Título.

CDU: 801

Glória Aparecida
Bibliotecária Documentalista - CRB-6/2047

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Estudos Linguísticos
Av. João Naves de Ávila, nº 2121, Bloco 1G, Sala 1G256 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4102/4355 - www.ileel.ufu.br/ppgel - secppgel@ileel.ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Estudos Linguísticos				
Defesa de:	Dissertação - PPGEL				
Data:	Vinte e oito de fevereiro de dois mil e vinte e três	Hora de início:	14:00	Hora de encerramento:	17:00
Matrícula do Discente:	12112ELI011				
Nome do Discente:	Heitor Carvalho de Almeida Neto				
Título do Trabalho:	GEConWeb: desenvolvimento de uma plataforma on-line para exploração de corpora				
Área de concentração:	Estudos em linguística e Linguística Aplicada				
Linha de pesquisa:	Teoria, descrição e análise linguistica				
Projeto de Pesquisa de vinculação:	Estudos em Linguística e Linguística Aplicada				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Estudos Linguísticos, assim composta: Professores Doutores: Vinícius Silva Pereira FAGEN/UFU; Maria Virgínia Dias de Ávila - FATRA; Ariel Novodvorski - UFU, orientador do candidato.

Iniciando os trabalhos, o presidente da mesa apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

APROVADO.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Ariel Novodvorski**, **Professor(a) do Magistério Superior**, em 28/02/2023, às 16:09, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Maria Virgínia Dias de Ávila**, **Usuário Externo**, em 28/02/2023, às 16:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Vinícius Silva Pereira**, **Professor(a) do Magistério Superior**, em 06/03/2023, às 10:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4288955** e o código CRC **4220AD01**.

Referência: Processo nº 23117.013726/2023-03 SEI nº 4288955

A Deus, meus pais e minha família.

A minha esposa. Maria do Socorro. e ao meu enteado, Pedro Henrique.

AGRADECIMENTOS

Lembrar de todos que merecem ser lembrados neste momento não é nada fácil. Por esse motivo, agradecerei aos envolvidos diretamente no processo que resultou neste trabalho final.

Primeiramente, agradeço a Deus por me ter conduzido até este momento, por ter me dado sabedoria e resiliência para poder continuar nesta jornada.

Ao Prof. Dr. Ariel Novodvorski, que aceitou o desafio de me guiar nessa jornada. Obrigado pela seriedade, leveza e sabedoria com que conduziu a orientação, demonstrando a confiança necessária para eu realizar a pesquisa e finalizar o curso. Mostrou-me que o ato da pesquisa pode ser feliz. Com muita admiração, serei sempre grato.

Aos meus pais, pelas palavras de encorajamento a cada desabafo que eu fazia em seus ouvidos.

Ao meu amigo, Antônio Machado, por ter me auxiliado na formatação do texto e pelas palavras de amparo em momentos de desabafo.

A minha companheira de vida, que me proporciona doses generosas e diárias de afeto e apoio em todos os aspectos, o que torna meus dias mais fáceis e felizes.

Gostaria de expressar minha gratidão ao PPGEL, ao ILEEL e à UFU pela oportunidade de crescer academicamente. Seu apoio foi fundamental para o meu desenvolvimento profissional, e estou muito grato por ter tido a oportunidade de estudar em uma instituição tão renomada. O conhecimento e as habilidades que adquiri durante meu tempo na UFU certamente me ajudarão em minha carreira futura.

RESUMO

A presente pesquisa tem como objetivo desenvolver uma plataforma *on-line* de análise exploratória e visualização de *corpora.* O intuito desse trabalho, além de dar visibilidade às pesquisas desenvolvidas pelos membros do grupo de pesquisa *GECon* (Grupo em Estudos Contrastivos), é disponibilizar para a comunidade acadêmica uma fonte de consulta, pesquisa e exploração de *corpora*, além de propiciar aos pesquisadores a possibilidade de divulgação de seus trabalhos, expondo os dados de suas pesquisas para outros pesquisadores e para a comunidade em geral. Essa é uma forma de dar notoriedade e visibilidade aos trabalhos desenvolvidos por esses e outros pesquisadores. A abordagem metodológica da Linguística de Corpus, em conjunto com os conceitos da ciência da computação, deu-nos os subsídios necessários para o desenvolvimento deste trabalho, possibilitando assim a construção de uma ferramenta linguística de análise de *corpora*. O resultado obtido com esse trabalho beneficiará não somente os membros do grupo *GECon*, mas a comunidade que se interessar pela consulta aos recursos, dispondo de uma fonte única de acesso a esses dados. A utilização dessa ferramenta poderá ser feita por meio da internet e ser acessada a partir de qualquer computador.

Palavras-chaves: *GECon*; Desenvolvimento da Plataforma; Linguística de Corpus; C*orpora*; Ciência da Computação.

Abstract

This research aims to develop an online platform for exploratory analysis and corpora visualization. The purpose of this work, in addition to giving visibility to the research carried out by the members of the research group *GECon* (Group of Contrastive Studies), is to provide the academic community with a source of consultation, research and manipulation of corpora, in addition to providing researchers with the possibility of disclosing their works exposing their research data to other researchers and to the community in general. This is a way of giving notoriety and visibility to the work developed by these and other researchers. The methodological approach of Corpus Linguistics, together with the concepts of computer science, gave us the necessary subsidies for the development of this work, thus enabling the construction of a linguistic tool for corpora analysis. The result obtained from this work will benefit not only the members of the *GECon* group, but an entire community itself, providing a single source of access to these data. Access to this tool can be done through the internet and can be accessed from any computer.

Keywords: *GECon*; Platform Development; Corpus Linguistics; Corpora; Computer Science.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

Esta pesquisa se encontra em fase final de desenvolvimento como parte obrigatória para a obtenção do título de mestre, no âmbito do Programa de Pós-Graduação em Estudos Linguísticos (PPGEL), do Instituto de Letras e Linguística (ILEEL), da Universidade Federal de Uberlândia (UFU), com vinculação à linha de pesquisa "Teoria, Descrição e Análise Linguística". O estudo tem como objetivo principal desenvolver uma plataforma que reunirá trabalhos produzidos pelos pesquisadores integrantes do grupo de pesquisa *GECon* (Grupo de Estudos Contrastivos), sob a orientação e a liderança do professor Dr. Ariel Novodvorski e de outros pesquisadores. A plataforma é constituída de ferramentas e recursos que permitem a exploração dos diferentes *corpora* que serão disponibilizados nela.

A ideia para a criação desta pesquisa surgiu quando o pesquisador estava trabalhando no desenvolvimento de uma ferramenta que permitia a visualização e a interação com um *corpus*. O trabalho em questão foi realizado para a pesquisa da doutoranda Ana Paula Corrêa Pimenta (2019), orientanda do professo Ariel, com o propósito de apresentar os vocábulos-termos do léxico sertanista, como um protótipo de vocabulário etnoterminológico *on-line*. A partir desse momento, o pesquisador começou a observar os professores e os alunos do ILEEL, instituto do qual faz parte como membro do corpo de servidores como técnico em informática, e notou que eles tinham dificuldade em divulgar, demonstrar e, até mesmo, disponibilizar de forma mais amigável o resultado dos seus trabalhos.

Uma das dificuldades identificadas que, portanto, carece de atenção, está na disponibilização dos *corpora*. Em razão disso, pensamos em uma maneira para que esses trabalhos pudessem ser visualizados de modo interativo e que fossem de fácil acesso para as pessoas. Foi pensando inicialmente em um *site web*[1], cujo acesso se dá por intermédio da *internet*, que nasceu a ideia deste trabalho. O site permitiria ao usuário selecionar qualquer uma das pesquisas disponibilizadas na plataforma e, assim, navegar e explorar os resultados obtidos por aquela pesquisa.

Desde que surgiu, em meados dos anos 2000, a *internet* vem passando por uma constante evolução, tornando-se o meio de comunicação mais difundido e utilizado nos últimos tempos. Essa rede de computadores tem sido utilizada, desde então, como o meio de maior acesso e disponibilização de informação. Sendo assim, mais e mais textos são

1 Termo comumente utilizado para se referir a *Word Wide Web*, rede mundial de computadores: *internet*.

disponibilizados diariamente nela, como: artigos científicos, obras literárias, notícias, vídeos, *e-mails* etc. Desse modo, não há motivos para que os trabalhos desenvolvidos pelos membros do grupo *GECon* não sejam também disponibilizados na *internet*, para que assim possam ser consultados por outros pesquisadores ou pessoas que se interessem por essa área do conhecimento.

Pesquisadores recorrem cada vez mais a esse meio de disponibilização de conteúdo e, com isso, obtêm um maior número de dados para as suas pesquisas, conseguindo assim maior rendimento e rapidez na seleção e na organização dessas informações. Conforme observado por Berber Sardinha (2004), para que seja possível o uso prático da Linguística de *Corpus* (LC), o pesquisador precisa de "um ingrediente essencial: o *corpus*" (BERBER SARDINHA, 2004, p. 45). Logo, a disponibilização de *corpora* na *internet* e em ferramentas de fácil acesso e navegação facilita sobremaneira a obtenção de dados oriundos de pesquisas já realizadas.

A utilização de computadores pessoais tem se tornado um recurso imprescindível para o estudo da LC. A aparição dos computadores e o avanço das tecnologias computacionais causaram uma transformação significativa na relação entre as diversas áreas de conhecimento e seus objetos de estudo. Atualmente, a capacidade do computador realizar atividades complexas e de armazenamento de informações também vêm influenciando nessa relação, uma vez que podemos obter e processar um número maior de dados. Berber Sardinha (2004, p. 17) afirma que o computador pessoal, com memória poderosa e capacidade de armazenamento, começa a desempenhar, nas ciências humanas, o papel transformador que o telescópio teve na física e nas ciências exatas.

O uso dos computadores ocorreu, pela primeira vez, na construção do *Corpus Brown University Standard Corpus of Present-Day American English*, que comumente tem sido chamado de *Brown Corpus*, que se tornou o primeiro *corpus* computadorizado de que se tem conhecimento. Esse *corpus* foi criado por Kucera e Francis e, apesar de pequeno, se comparado aos padrões atuais, alcançou a marca de um milhão de palavras. Se levarmos em conta as condições tecnológicas da época, em que a utilização de computadores se restringia a poucos centros universitários e sua capacidade de processamento era pequena, uma vez que o armazenamento dos dados era feito com cartões perfurados, esse é um número elevado de palavras.

Berber Sardinha (2004, p.20) ressalta que são necessários quatro pré-requisitos para a constituição de um *corpus*:

> 1. 0 corpus deve ser composto de textos autênticos, em linguagem natural. assim, os textos não podem ter sido produzidos com o propósito de serem alvo de pesquisa linguística, e não podem ter sido criados em linguagem artificial, tal como linguagem de programação de computadores ou notação matemática.
> 2. Autenticidade dos textos subentende textos escritos por falantes nativos. Tanto assim que, quando esse não é o caso, deve-se qualificá-lo como corpora de aprendizes (*learner corpora*).
> 3. O conteúdo do corpus deve ser escolhido criteriosamente. Os princípios da escolha dos textos devem seguir, acima de tudo, as condições de naturalidade e autenticidade. Mas devem também obedecer a um conjunto de regras estabelecidas por seus criadores de modo que o corpus coletado corresponda às características desejadas. Por exemplo, se é um corpus de português brasileiro escrito que represente a língua portuguesa, tal qual é escrita no Brasil, em sua totalidade, a coleta deve ser guiada por um conjunto de critérios que garanta, entre outras coisas, que o maior número possível de tipos textuais existentes no português brasileiro esteja representado, que haja uma quantidade aceitável de cada tipo de texto e que a seleção dos textos seja aleatória, a fim de não contaminar a coleta com variáveis indesejáveis.
> 4. Representatividade. Tradicionalmente, tende-se a ver um corpus como um conjunto representativo de uma variedade linguística ou mesmo de um idioma. Mas a questão não pode ser enfocada no vácuo. Cabe perguntar: representativo do quê e para quem? A representatividade será discuti da com mais detalhes.

Segundo dois renomados linguistas da área, o uso do termo *corpus* tem implicações bastante específicas. Segundo McEnery e Wilson (1996), a moderna noção de *corpus* carrega consigo características fundamentais como:

> - formato eletrônico (*machine-readable form*): atualmente o emprego do termo corpus significa admitir necessariamente que os textos estejam no formato eletrônico, diferentemente da ideia que se tinha de corpus no passado, a qual se referia somente a textos impressos. Ainda de acordo com McEnery e Wilson (1996), o formato possui vantagens consideráveis: i) os corpora podem ser pesquisados e manipulados de forma mais rápida; ii) os corpora podem ser mais facilmente enriquecidos com informação extra;
> - referência padrão (*standard reference*): ainda de acordo com McEnery e Wilson (1996), existe um entendimento de que um corpus constitui uma referência padrão para a variedade de língua que ele representa, pressupondo que o corpus esteja disponível para outros pesquisadores, em outras palavras, é o que se tem chamado de reuso do corpus. (MCENERY; WILSON, 1996, p. 17)

Entre essas duas características ressaltadas pelos autores, precisamos destacar a referência padrão (*standard referecence*), em que é dada uma importância não só à forma como o *corpus* é disponibilizado, mas também que todo o esforço não seja direcionado apenas para sua construção. É necessário que também nos esforcemos para deixá-lo acessível para outros pesquisadores.

Foi nesse sentindo que pensamos no desenvolvimento deste trabalho, em deixar os *corpora* disponíveis e acessíveis para que os eventuais interessados possam consumi-los e manipulá-los. Identificamos aí uma lacuna, que ainda necessita de atenção, e essa

demanda está relacionada à disponibilidade de ferramentas computacionais direcionadas à publicação, visualização e manipulação de *corpora* que possibilitem a obtenção de resultados a partir dos *corpora* de estudo de outros pesquisadores.

O meio eletrônico se tornou um dos meios de comunicação mais importantes da atualidade, pois grande parte das obras literárias possui pelo menos uma versão on-line. Sendo assim, optamos por desenvolver uma plataforma também on-line que permitirá a visualização e a manipulação de *corpus*. Por intermédio dessa plataforma, o usuário poderá visualizar e manipular o *corpus* de uma forma interativa e, com isso, vislumbrar de forma prática e rápida os resultados a partir dos *corpora* disponibilizados pela plataforma.

Encontramos, portanto, uma forma de ajudar os membros do grupo de pesquisa *GECon*. Essa plataforma possibilitará aos pesquisadores da área em LC disponibilizar seus trabalhos e, ao mesmo tempo, consumir os trabalhos desenvolvidos por outros pesquisadores. E para o público em geral, será um mecanismo de consulta e de pesquisa, onde poderão conhecer os trabalhos que estão sendo desenvolvidos nessa área.

Diante do contexto apresentado, surgira alguns questionamentos que nortearam o desenvolvimento desta pesquisa: i) Como integrar os recursos necessários para o desenvolvimento de pesquisas que se servem da exploração de *corpora*? e ii) De que modo viabilizar o acesso aos recursos mais relevantes e de maior procura e utilização por pesquisadores da LC e da comunidade em geral, numa plataforma linguística?

Em razão do exposto anteriormente, na próxima seção apresentaremos o objetivo geral e os objetivos específicos que direcionaram o desenvolvimento desta pesquisa.

1.1 OBJETIVOS

A seguir, apresentamos os objetivos que nortearam a pesquisa, e que têm como base os conceitos da Ciência da Computação acerca do desenvolvimento de *software*, e as definições teórico-metodológicas da LC.

1.1.1 Objetivo geral

1) Projetar o desenvolvimento de uma plataforma baseada na utilização de diferentes *corpora*, que permitirá aos interessados em estudos linguísticos e literários, docentes, pesquisadores e/ou estudantes, explorarem, por via remota, os *corpora* estudados por outros pesquisadores e disponibilizados por intermédio da plataforma.

1.1.2 Objetivos específicos

1) Desenvolver uma plataforma, cuja interface seja interativa e intuitiva, para exploração de *corpora*.
2) Testar e ajustar as funcionalidades da plataforma com os *corpora* compilados e durante seu desenvolvimento, no intuito de conferir a extração de dados que possibilite diferentes tipos de análises linguísticas.
3) Disponibilizar para a comunidade acadêmica e interessados em geral a consulta aos dados decorrentes de pesquisas do *GECon*, possibilitando uma divulgação mais ampla.

Convém esclarecer que, por intermédio do desenvolvimento dessa plataforma, pretendemos dar mais visibilidade e notoriedade às pesquisas desenvolvidas por membros do grupo de pesquisa *GECon* e, com isso, possibilitar e incentivar o desenvolvimento de mais trabalhos nessa área do conhecimento.

Além da Introdução, esta dissertação está organizada em três capítulos. Na Fundamentação Teórico-metodológica (Capítulo 2), apresentamos os preceitos teóricos e metodológicos deste trabalho e algumas ferramentas que nos permitiram a manipulação de *corpora*. No Capítulo 3, tratamos sobre o *corpus* e sobre as metodologias que foram utilizadas para a construção de uma plataforma e compilação de um *corpus*. Por fim, no Capítulo 4, apresentamos as ferramentas que estão disponíveis na plataforma. A seguir, discutimos os principais conceitos da LC, os quais nortearam o desenvolvimento deste trabalho.

FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA

Neste capítulo, apresentamos o referencial teórico e metodológico no qual se fundamenta nossa pesquisa. Para isso, dividimos o capítulo, em subseções, nas quais discorremos inicialmente sobre a Linguística de *Corpus*. Em seguida, estabelecemos a relação entre Ciências da Computação e Linguística de *Corpus*; posteriormente, tratamos da Linguística de *Corpus* e das ferramentas básicas utilizadas para análise de um *corpus*.

2.1 CIÊNCIAS DA COMPUTAÇÃO E LINGUÍSTICA DE *CORPUS*

Os *corpora* têm sido empregados como fontes de pesquisa em variadas áreas do conhecimento, incluindo a Tradução, a Lexicologia, a Lexicografia, a Terminologia e a Terminografia, além das subáreas da Linguística. De acordo com Tagnin (2015, p. 38), os *corpora* têm sido utilizados em investigações que envolvem a comparação de textos originais e suas traduções para buscar equivalentes. A Fraseologia é outra área que se desenvolveu com o ferramental metodológico e teórico disponibilizado pela LC, que permite identificar recorrências de agrupamentos lexicais com muita facilidade.

A computação é uma das áreas do conhecimento que também tem recorrido à LC como forma de obtenção de dados para o aperfeiçoamento de algoritmos de Processamento de Linguagem Natural - PLN (PUSTEJOVSKY; STUBBS, 2012, p. 13). O Processamento de Linguagem Natural é uma subárea da Inteligência Artificial e da Linguística Computacional que se dedica a estudar a interação entre computadores e a linguagem humana. O objetivo do PLN é permitir que as máquinas entendam, processem e gerem a linguagem natural humana de forma semelhante aos seres humanos, incluindo a capacidade de reconhecer a fala, processar textos, responder perguntas e produzir textos em linguagem natural. As aplicações do PLN são diversas, como *chatbots*, sistemas de recomendação, análise de sentimento, tradução automática, entre outras.

Outra área que também tem se beneficiado dos trabalhos e da evolução da LC é a Tradução, principalmente a tradução técnica.

Apesar dessa gama de áreas do conhecimento que se valem da LC como ingrediente principal para o desenvolvimento e o aprimoramento de suas pesquisas, identificamos uma lacuna que ainda necessita de atenção, e essa demanda está relacionada à disponibilidade de ferramentas computacionais direcionadas à visualização e à exploração de *corpora*, a fim de obter resultados a partir dos *corpora* de estudo de outros pesquisadores.

A seguir, mostramos algumas ferramentas que foram desenvolvidas com o intuito de maximizar determinadas atividades da LC, e que foram criadas com o intuito de facilitar a vida dos linguistas.

2.2 LINGUAGEM DE PROGRAMAÇÃO *PYTHON*

Assim como existem várias línguas naturais (inglês, francês, espanhol etc.), com diferentes características, pronúncias e sintaxes variadas, temos também várias linguagens artificiais utilizadas como meio de nos comunicar com o computador, que são chamadas de linguagens de programação como, por exemplo, C++, Java, C#[2], etc. As linguagens de programação possuem também características e sintaxe variadas. Ser proficiente em várias línguas naturais não é tarefa simples, do mesmo modo acontece com as linguagens de programação. Dominar características, peculiaridades e sintaxe em diversas linguagens não é algo tão trivial.

Abordamos, a seguir, alguns dos principais aspectos, funcionalidades e características da linguagem de programação *Python*[3]. Trata-se de uma linguagem de programação muito poderosa e versátil. Embora não seja objeto de nosso trabalho detalhar todos os recursos e conceitos da linguagem, apresentamos algumas de suas características. Por meio dela podemos desenvolver desde grandes sistemas até ferramentas para automatização de tarefas, *sites* de comércio eletrônico, ferramentas de análise de dados e para análise linguística.

Python é uma linguagem de alto nível, desenhada para ser usada de uma forma ampla e geral, ou seja, com ela conseguimos resolver diversos tipos de problemas em várias áreas, seu alto nível lhe confere uma facilidade extra na hora do aprendizado. Seu poder computacional além de uma sintaxe concisa, enxuta e clara facilitam e contribuem para sua escolha, quando vamos criar projetos.

Recursos poderosos que podem ser facilmente adicionados por meio de uma vasta lista de bibliotecas fazem com que ela seja utilizada em vários cenários, desde a educação, até projetos que fazem uso de inteligência artificial. *Python* vem se tornando uma das

2 C# pronuncia-se C Sharp, é uma linguagem de programação orientada a objetos desenvolvida pela Microsoft. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>. Acesso em: 10 nov 2022.

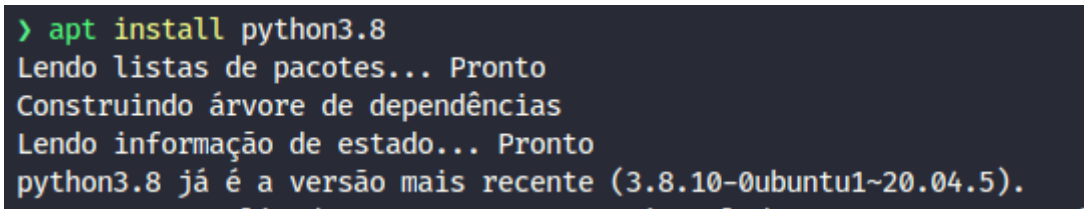
3 *Python* é uma linguagem que foi desenvolvida em 1991 pelo cientista da computação Guido Van Rossumem, com o intuído de ser simples e de fácil compreensão. Disponível em: <https://pt.wikipedia.org/wiki/Python>. Acesso em: 04 nov 2022.

linguagens mais utilizadas no mercado de desenvolvimento de *software* devido a essas características. Dentre a variedade de bibliotecas que estão disponibilizadas e que podem ser utilizadas com o *python*, temos algumas que se destacam quando se trata de carregamento e manipulação de dados, da geração e visualização (gráficos), análise estatística e do processamento de linguagem natural. Essas bibliotecas conferem à linguagem um poder de resolução de problemas muito vasto, sendo em sua grande maioria bibliotecas gratuitas e de código fonte aberto (*Open Source*[4]) e que recebem atualizações periodicamente.

Para fazer o gerenciamento dessas bibliotecas no *python,* utilizamos uma ferramenta chamada pip*,* um gerenciador de pacotes por meio do qual é possível controlar as versões dos pacotes que serão instalados ou que já estão instalados no nosso projeto de *software*. Para instalar esse gerenciador de pacotes temos que fazer o *download* do arquivo de instalação e, após realizar a instalação, conforme ilustrado na Figura 2. Antes mesmo de instalar o pip, temos que instalar o *python* ou nos certificarmos de que ele já esteja instalado em nosso sistema operacional.

Como o *python* é uma linguagem que vem ganhando muitos adeptos, em alguns sistemas operacionais, principalmente aqueles baseados na arquitetura *Unix,* como *Linux* e *MacOS*, já vem instalado por padrão. Entretanto, caso não tenha sido instalado, essa instalação pode ser feita executando o comando abaixo:

Figura 1 Comando para instalar o python em sistemas operacionais unix



Fonte: O autor

Para usuário do sistema operacional *Windows* é necessário fazer a instalação manualmente do *python*. Para instalá-lo no *Windows* precisamos fazer o *download* do arquivo de instalação, que pode ser encontrado no *site* oficial: https://www.python.org/downloads/. O processo de instalação é bem simples, basta ir clicando em *next* (próximo) nas telas que aparecem durante o processo de instalação. Caso

4 *Open source* é um termo que se refere a software livre; um software livre ou *open source* pode ser acessado abertamente pelo público: todas as pessoas podem vê-lo, modificá-lo e distribuí-lo conforme suas necessidades. Disponível em: <https://fia.com.br/blog/open-source/>. Acesso em: 19 out 2022.

o *python* não esteja instalado no computador, não será possível realizar a instalação do gerenciador de pacotes pip.

Figura 2 Instalando o gerenciado de pacotes pip

```
❯ sudo python3 get-pip.py
Collecting pip
  Downloading pip-22.2.2-py3-none-any.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 10.0 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
Successfully installed pip-22.2.2
```

Fonte: O autor

Pacote é um conjunto de arquivos que são necessários para que um módulo funcione; módulos são um conjunto de códigos em *python*, que foram criados para resolver determinado problema. A figura ilustra a estrutura de um pacote/biblioteca.

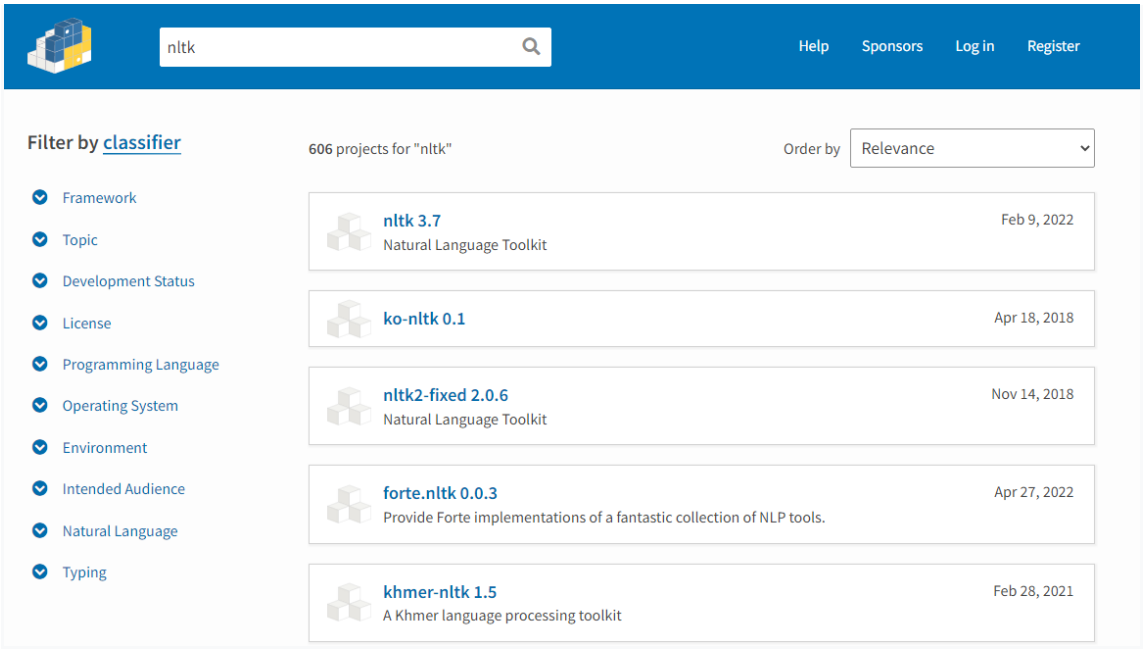
Figura 3 Estrutura de uma biblioteca em *python*

.github/workflows	Include a hash of third-party.sh in the third party tools cache	16 days ago
nltk	Merge pull request #3054 from tomaarsen/tests/doctests	10 days ago
tools	Update Stanford CoreNLP version for CI tests	16 days ago
web	Blackified conf.py	4 months ago
.gitattributes	Introduce end-of-line normalization	10 years ago
.gitignore	Add .DS_Store to .gitignore for macOS users	7 months ago
.pre-commit-config.yaml	Update black to 22.3.0	7 months ago
AUTHORS.md	Docstring tests (#3050)	22 days ago
CITATION.cff	Add `CITATION.cff` to nltk (#2880)	11 months ago
CONTRIBUTING.md	Drop support for Python 3.6, support Python 3.10 (#2920)	10 months ago
ChangeLog	Prepare for NLTK 3.7	8 months ago
LICENSE.txt	Use the full license text and a separate notice file	2 years ago

Fonte: <https://github.com/nltk/nltk>

No repositório central do pip temos a nossa disposição milhares de pacotes, com os quais podemos contar para solucionar os problemas da nossa aplicação. Estão catalogados no *site* 406.764 projetos, 3.853.622 lançamentos, 6.871.951 arquivos e 629.393 usuários. Seria uma tarefa impossível localizar uma biblioteca no meio de tantos outros, mas para encontrar um pacote podemos utilizar o campo de pesquisa, bastando somente digitar o nome do pacote que queremos que o *site* listará todos os que forem compatíveis com o termo digitado.

Figura *4* Pesquisando por um pacote no repositório central do *pip*.



Fonte: https://pypi.org/search/?q=nltk

Para instalar um pacote/biblioteca em um projeto, devemos executar o seguinte comando: *pip install nome_do_pacote* como, por exemplo: *pip install nltk*, conforme ilustrado pela Figura 5. Neste caso em específico, estamos instalando o pacote chamado Natural Language Toolkit *NLTK*.

Figura 5 Instalando o pacote *NLTK* com pip

```
❯ via exemple_pip python3 -m pip install nltk
Collecting nltk
  Downloading nltk-3.7-py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 3.7 MB/s
Collecting tqdm
  Downloading tqdm-4.64.1-py2.py3-none-any.whl (78 kB)
     |████████████████████████████████| 78 kB 7.2 MB/s
Collecting regex>=2021.8.3
  Downloading regex-2022.9.13-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (772 kB)
     |████████████████████████████████| 772 kB 12.2 MB/s
Collecting joblib
  Downloading joblib-1.2.0-py3-none-any.whl (297 kB)
     |████████████████████████████████| 297 kB 11.1 MB/s
Collecting click
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     |████████████████████████████████| 96 kB 6.3 MB/s
Installing collected packages: tqdm, regex, joblib, click, nltk
Successfully installed click-8.1.3 joblib-1.2.0 nltk-3.7 regex-2022.9.13 tqdm-4.64.1
```

Fonte: O autor

Além das características citadas e das facilidades que a linguagem nos fornece, a linguagem *Python* é multiplataforma, o que significa dizer que os programas criados por ela podem ser executados nos principais sistemas operacionais da atualidade (*Linux, Windows e MacOS*). Essa característica da linguagem é muito interessante pois, uma vez

desenvolvido, o programa poderá ser transportado para qualquer computador que tenha instalado um dos sistemas operacionais listados anteriormente.

Sendo assim, com características tão peculiares e com tantas bibliotecas disponíveis para manipulação de textos e extração de conteúdo da *internet*, a linguagem de programação *Python* nos pareceu ser a escolha mais natural e que mais se adequa ao objetivo que queremos alcançar com o desenvolvimento desta pesquisa, pois nos permitirá a realização de análise linguística, a partir das obras compiladas nele.

A linguagem de programação *python* foi utilizada como motor para a exploração dos dados e para sua extração; contudo, precisávamos de um meio para expor essas informações e, para realizar essa tarefa, tivemos que escolher uma segunda linguagem de programação que nos fornecesse meios para tal. Falaremos um pouco da linguagem de programação Java e dos motivos pelos quais foi escolhida para essa finalidade.

2.3 LINGUAGEM DE PROGRAMAÇÃO JAVA

O Java[5] foi desenvolvido na década de 90 por um conjunto de engenheiros de *software*, liderados por James Gosling, na empresa Sun Microsystems. O Java é uma das linguagens de programação mais utilizadas no mundo, tendo a sua primeira versão liberada para a comunidade em 1995, pela Sun Microsystems. Acreditando que haveria uma convergência dos computadores, dos equipamentos e eletrodomésticos, a empresa desenvolveu uma linguagem de programação que poderia ser facilmente executada em diferentes meios e equipamento, podendo até ser executada em redes como a *internet*. Com essa nova linguagem, as aplicações poderiam ser executadas dentro dos navegadores nos chamados *Applets Java* [6] e tudo seria disponibilizado pela *internet* instantaneamente por meio de navegadores web.

Assim surgiu o Java, uma linguagem de programação voltada para o desenvolvimento de ferramentas a, serem disponibilizadas na *internet* e consumidas por diversos tipos de equipamentos, desde uma TV, micro-ondas até um dispositivo móvel como o celular.

---

5 Definição para a linguagem de programação Java https://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o). Acessado em: 01 nov. 2022:

6 O termo "applet" é uma combinação das palavras "aplicativo" e "pequeno" (em inglês, "app" e "let"), pois esses programas são pequenos e projetados para executar tarefas específicas.

Essa ideia deu certo e a linguagem foi crescendo e se desenvolvendo cada vez mais, tanto que, hoje as empresas utilizam para o desenvolvimento de diversas aplicações, tanto WEB como Mobile (para celular), tornando seu uso difundido em todo o mundo.

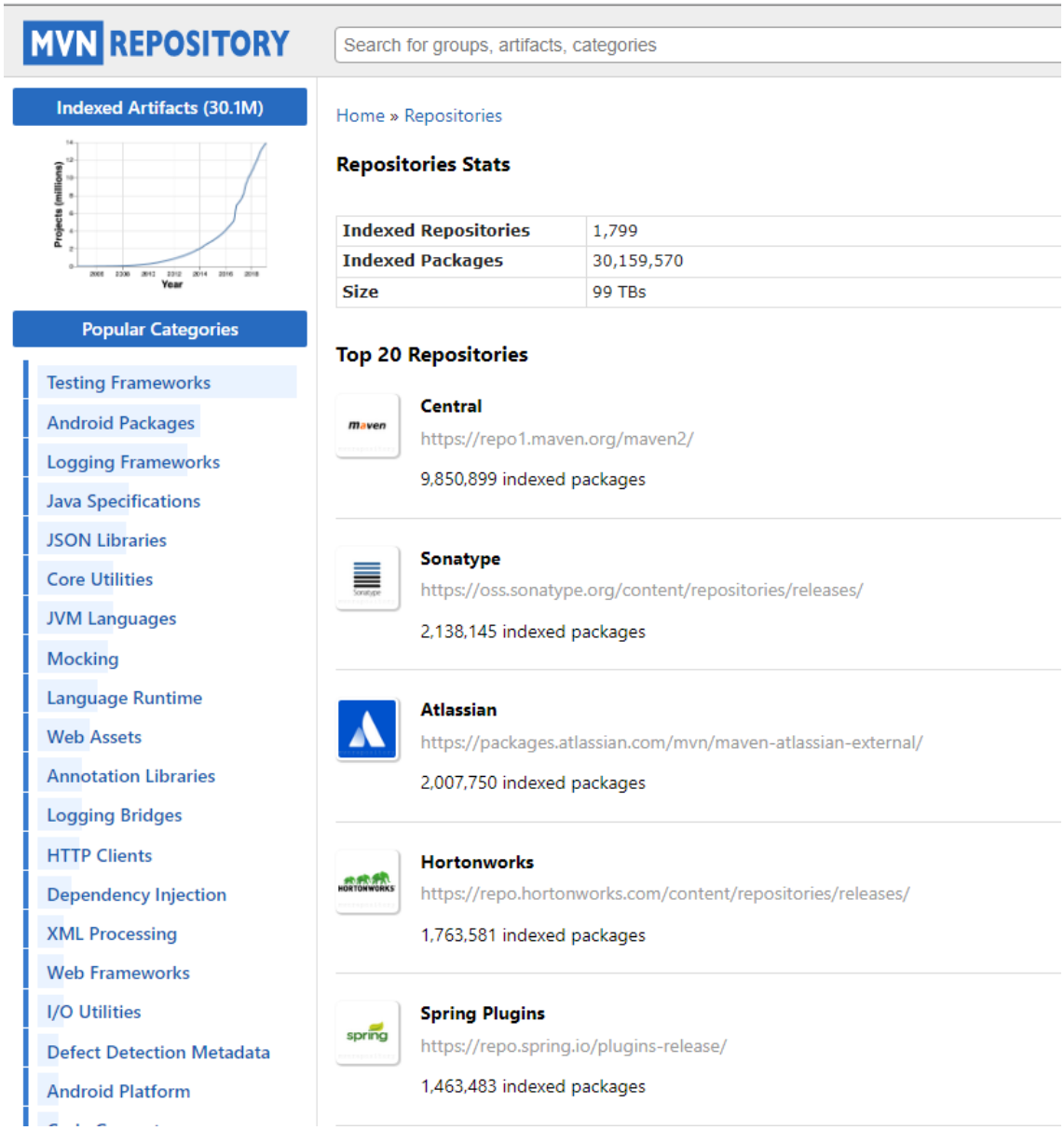
Assim como em Python, existem várias bibliotecas para serem utilizadas com Java. Em Python utilizamos o gerenciador de pacotes chamados pip; já em Java utilizamos o nvm, abreviação de Apache Maven. O Maven ou Apache Maven é um *software* de código fonte livre mantido pela fundação Apache[7]. Trata-se de uma ferramenta de gestão de dependências, e diferentemente do pip, o maven vai além do gerenciamento das bibliotecas que são instaladas no nosso projeto de *software,* ele também é um *task runner*[8]; em outras palavras, o Maven além de fazer a gerência das bibliotecas também automatiza os processos de obtenção de dependências e compilação de projetos em Java, podendo, além dessas, executar outras tarefas, como testes unitários de integração etc.

É comum, no processo de desenvolvimento de um software, utilizarmos diversas bibliotecas e cada uma delas com versões diferentes, que podem ser incompatíveis entre si; por isso, temos a necessidade de uma ferramenta que faça a gerência e que não deixe que ocorram conflitos entre diferentes versões de uma mesma biblioteca.

O Maven dispõe, assim como no pip, de um repositório central ver ilustração da **Figura** 6, em que são disponibilizadas as bibliotecas com as quais queremos trabalhar. O processo de obtenção de uma biblioteca em específico é um pouco diferente do utilizado pelo pip no *Python*, e caso já saibamos o nome da biblioteca podemos adicioná-lo juntamente com a versão e o nome da organização que disponibilizou essa biblioteca em um arquivo chamado pom.xml[9].

7 Apache é uma organização sem fins lucrativos criada para suportar os projetos de código aberto. Disponível em: https://mvnrepository.com/artifact/org.apache.maven. Acessado em: 04 nov 2022.
8 Gerenciador de tarefas, organiza a execução dos processos de compilação de um software.
9 POM é um acrônimo para *Project Object Model*. Disponível em: https://www.javatpoint.com/maven-pom-xml. Acessado em: 04 nov 2022.

Figura 6 Repositório central do *maven*



Fonte: https://mvnrepository.com/

O arquivo pom.xml contém informações do projeto e de configuração para o Maven construir o projeto, como dependências, diretório de compilação, diretório de origem, diretório de origem de teste etc.

O Maven lê o arquivo pom.xml a fim de executar as tarefas que estão contidas nele, seja para executar testes ou para baixar bibliotecas ou até para compilar o projeto, conforme a **Figura** *7*, que ilustra um arquivo de configuração do Maven.

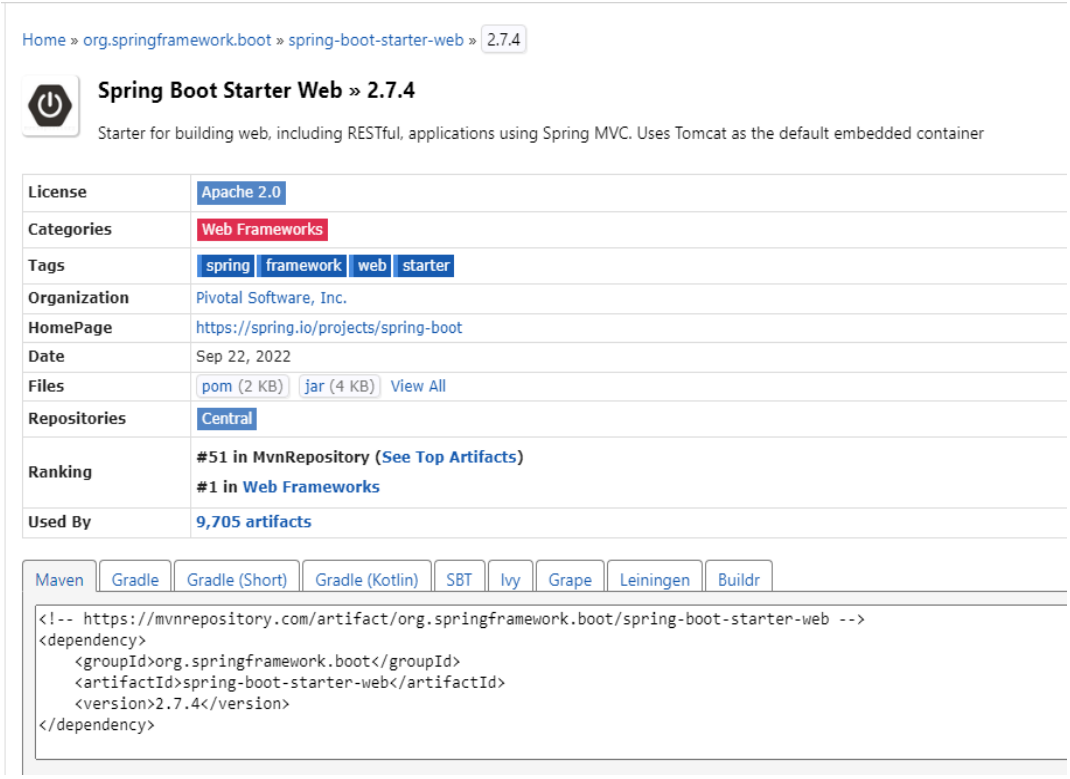
Figura 7 Arquivo de configuração do maven - POM

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>4.3.4.RELEASE</version>
</dependency>

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.1.0</version>
  <scope>provided</scope>
</dependency>

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

Fonte: O autor

Todos os dados da biblioteca, a serem instalados no projeto, podem ser acessados no *site* do repositório central, essas informações podem ser encontradas facilmente. Buscando pelo nome da biblioteca no campo de pesquisa do *site,* será retornada uma lista com os itens que coincidiram com o termo da pesquisa. Após clicar em um deles, os detalhes dessa biblioteca serão exibidos, conforme ilustra a **Figura** *8*. De posse dessas informações, bastará adicioná-las em arquivos *pom.xml,* que ele irá automaticamente baixar e instalar a biblioteca no projeto, deixando-a pronta para ser utilizada.

Figura 8 Pesquisando por uma biblioteca no repositório central do maven



Fonte: https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web/2.7.4

O Java foi escolhido para fornecer uma interface *web* para acesso aos dados de nosso trabalho, devido ao grande número de *frameworks*[10] disponíveis no mercado que facilitam e agilizam o desenvolvimento de páginas *web*.

2.3.1 Diferença entre Bibliotecas e *Frameworks*

Um *framework* consiste, basicamente, em elementos de código que guiam o desenvolvimento de uma aplicação, otimizando esse processo. Por causa disso, às vezes pode ser confundido com uma biblioteca. Mas eses são dois conceitos distintos, utilizados para alcançar objetivos diferentes. Um *framework* representa a estrutura dentro da qual será desenvolvido um *software*. Assim, o código deve — desde o princípio — seguir os padrões estabelecidos pelo *framework*. Já uma biblioteca representa recursos que poderão ser utilizados no decorrer do desenvolvimento, fornecendo elementos para completar uma etapa do desenvolvimento ou otimizá-lo.

2.3.2 Por que precisamos de framework?

10 São conjuntos de ferramentas, bibliotecas e padrões de codificação que fornecem uma estrutura de suporte para o desenvolvimento de *software*. Disponível em https://kenzie.com.br/blog/framework/. Acesso em: 04 nov. 2022.

Os *frameworks* oferecem facilidades e flexibilidade para trabalharmos quando estamos desenvolvendo aplicações; para qualquer linguagem que tenhamos escolhido, queremos nos preocupar o mínimo possível com questões relacionadas à infraestrutura.

O *Spring* traz para o desenvolvimento *web* uma camada extra de abstração e facilitação, com o objetivo de fornecer ao programador uma facilidade extra nas configurações da aplicação, deixando-o livre para focar na lógica de negócio da aplicação.

Essas são algumas das facilidades que podemos obter com o *Spring*, fazendo dele uma excelente escolha para o nosso trabalho. Por esses e outros motivos, optamos por utilizar o *Spring* no desenvolvimento das páginas *web* de nossa pesquisa.

Em seções que se seguirão, trataremos da parte metodológica para a criação e compilação de um *corpus,* dos detalhes e procedimentos que devemos seguir para a compilação dos textos que serão utilizados por ferramentas de análise linguística. Na próxima seção, daremos mais detalhes acerca da LC e dos *corpora*.

2.4 *CORPUS* E LINGUÍSTICA DE *CORPUS*

O trabalho aqui proposto tem como fundamentação teórica os preceitos da LC. De acordo com Berber Sardinha (2004, p. 3), a LC "pode ser definida como uma área que se ocupa da coleta e exploração de *corpora*, ou conjunto de dados linguísticos textuais coletados criteriosamente, com o propósito de servirem para a pesquisa de uma língua ou variedade linguística".

Ainda no âmbito da LC, para que um conjunto de dados seja considerado um *corpus*, existem alguns princípios e critérios que devem ser levados em consideração. Segundo Berber Sardinha (2004, p.18-19), são eles:

- a origem: pois os dados devem ser autênticos;
- o propósito: os dados devem ser objeto de estudo linguístico;
- a composição: os dados devem ser escolhidos com critério;
- a formatação: os dados devem ser legíveis por computadores;
- a representatividade: os dados devem ser representativos de uma língua, de um idioma, ou de uma variedade deles;
- a extensão: o *corpus* deve ser vasto para ser representativo.

Tomando por base a definição dada por McEnery e Wilson (1996), a moderna noção de *corpus* carrega consigo pelo menos quatro características fundamentais:

> a) amostragem e representatividade (*sampling and representativeness*): um *corpus* deve ter uma amostragem suficiente da língua ou variedade de língua que se quer analisar para obter se o máximo de representatividade desta mesma língua ou variedade de língua;
> b) tamanho finito (*finite size*): com exceção de corpus-monitor[11], todo corpus tem um tamanho finito, por exemplo: 500 mil palavras, 1 milhão de palavras, 10 milhões de palavras etc.
> c) formato eletrônico (*machine-readable format*): segundo McEnery e Wilson (1996), atualmente o emprego do termo *corpus* significa admitir necessariamente que os textos estejam no formato eletrônico, diferentemente da ideia que se tinha de corpus no passado, a qual se referia somente a textos impressos. Ainda de acordo com esses autores, o formato eletrônico possui vantagens consideráveis: i) os corpora podem ser pesquisados e manipulados de forma mais fácil e rápida; ii) os corpora podem ser mais facilmente enriquecidos com informação extra;
> d) referência padrão (standard reference): ainda de acordo com McEnery e Wilson (1996), existe um entendimento tácito de que um *corpus* constitui uma referência padrão para a variedade de língua que ele representa, pressupondo que o *corpus* esteja disponível para outros pesquisadores, em outras palavras, é o que se tem chamado de reuso do corpus.

Para Galisson e Coste (1983, p.763), *corpus* é:

> um conjunto finito de enunciados tomados como objeto de análise. Mais precisamente, conjunto finito de enunciados considerados característicos do tipo de língua a estudar, reunidos para servirem de base à descrição e, eventualmente, à elaboração de um modelo explicativo dessa língua. Trata-se, pois, de uma coleção de documentos quer orais (gravados ou transcritos) quer escritos, quer orais e escritos, de acordo com o tipo de investigação pretendido. As dimensões do corpus variam segundo os objetivos do investigador e o volume dos enunciados considerados como característicos do fenômeno a estudar. Um corpus é chamado exaustivo quando compreende todos os enunciados característicos. É chamado seletivo quando compreende apenas uma parte desses enunciados.

Biderman (2001, p. 79) fornece, ainda, uma segunda concepção de *corpus*:

> Pode-se definir um *corpus* linguístico informatizado assim: é uma coletânea de textos selecionados segundo critérios linguísticos, codificados de modo padronizado e homogêneo. Essa coletânea pode ser tratada mediante processos informáticos.

Para Dubois et al. (1993, p.653), *corpus* é considerado o conjunto de enunciados a partir do qual se estabelece a gramática descritiva de uma língua:

11 Corpus-monitor é um *corpus* que recebe o incremento de novos textos e se tornar maior à medida que os textos são inseridos nele.

> *corpus* não pode ser considerado como constituindo a língua, mas somente como uma amostra da língua. (...) O *corpus* deve ser representativo, isto é, deve ilustrar toda a gama das características estruturais. Poder-se-ia pensar que as dificuldades serão levantadas se um *corpus* for exaustivo (...). Na realidade, sendo indefinido o número de enunciados possíveis, não há exaustividade verdadeira e, além disso, grandes quantidades de dados inúteis só podem complicar a pesquisa, tornando-a pesada. O linguista deve, pois, procurar obter um *corpus* realmente significativo. Enfim, o linguista deve desconfiar de tudo o que pode tornar o seu *corpus* não representativo (método de pesquisa escolhido, anomalia que constitui a intrusão de linguista, preconceito sobre a língua).

Além de atentar para a relação de tamanho do *corpus*, Berber Sardinha (2004) expõe que um *corpus* deve ser o mais extenso possível:

> O *corpus* é uma amostra de uma população cuja dimensão não se conhece (a linguagem como um todo). Desse modo, não se pode estabelecer qual seria o tamanho ideal da amostra para que represente essa população. Uma salvaguarda é tornar a amostra a maior possível, a fim de que ela se aproxime ao máximo da população da qual deriva, sendo, portanto, mais representativa (BERBER SARDINHA, 2004, p.23).

Entretanto, Fromm (2003, p. 8) ressalta que o desenvolvimento de um *corpus* muito grande pode requerer a participação de vários pesquisadores e/ou auxiliares. Do contrário, a sua construção pode consumir muito tempo ou até anos para que ele seja concluído.

Nesses casos, a questão é o tempo disponível que o pesquisador (ou equipe de pesquisadores) tem para se dedicar à obtenção de dados. Berber Sardinha (1999) salienta que, na prática, "o pesquisador coleta certa quantidade de dados de acordo com suas possibilidades, efetua a análise, mas não sabe se sua coleta foi além ou aquém do que seria teoricamente mais adequado" (BERBER SARDINHA, 1999, p. 4). Por essa razão, Nelson (2010, p. 30) afirma que a criação de um *corpus* é "uma aceitação entre o que é esperado e o que é possível"[12]

Mesmo com as recomendações e ressalvas dos autores citados sobre a atenção que devemos dar quanto ao tamanho que um *corpus* deve ter e sobre o tempo que demanda para poder obter e processar esses dados, devemos ter em mente que esses dados podem não ser suficientes ou podem até mesmo extrapolar a quantidade essencial demanda para aquele *corpus*. Hoje, diferentemente do que acontecia no passado, temos

12 Texto original: "any attempt at corpus creation is therefore a compromise between the hoped for and the achievable".

cada vez mais dados disponíveis na *internet* e os progressos tecnológicos para o processamento dos mesmos melhoraram de forma a agilizar e a facilitar essa coleta. Sendo assim, diferente do que ocorria no passado não precisamos temer e nos restringir a uma única fonte de obtenção dos dados. Como a extensão de um *corpus* está sujeita à disponibilidade dos dados, hoje podemos obter uma grande quantidade de dados com uma certa facilidade. As precauções que pesquisadores tinham no passado se justificavam pelas limitações tecnológicas da época, agora isso pode não ser mais motivo para nos preocuparmos.

Ao estudarmos os conceitos apresentados anteriormente, defendidos por autores renomados da área, podemos chegar à conclusão de que, para a construção de um *corpus,* devemos levar em consideração aspectos importantes, como a representatividade do mesmo, ou seja, quanto maior for o número de termos, mais representativo ele será. Além disso, os textos devem ser escritos de forma natural, por falantes nativos da língua, isto é, não podem ser produzidos para a pesquisa em específico e deve ser possível, a partir desses textos, conseguir ampliar o conhecimento das estruturas da língua que ele representa.

É importante salientar que o uso do computador e das tecnologias computacionais tem um papel primordial no desenvolvimento da linguística de *corpus*, entretanto, não podemos deixar de lado as grandes contribuições que os *corpora* compilados manualmente deram para a linguística.

Isso vem demonstrar quão necessária e importante é a utilização do computador para a realização de trabalhos na área da LC.

Conforme defendido por Berber Sardinha (2004, p.18-19), um *corpus* deve ser extenso e representativo, além disso o pesquisador deve se atentar para a relação de tamanho dos *corpora*. Considerando essa premissa, é difícil imaginar uma maneira de analisar grandes quantidades de textos sem o uso de ferramentas computacionais, já que a análise manual de volumes extensos de textos se tornou impraticável nos dias de hoje.

O computador e suas ferramentas são de grande relevância para a área da linguística, como evidenciado em outros trabalhos da área. A Computação nos forneceu fundamentos teóricos e metodológicos para a produção da plataforma *GEConWeb*[13]. Trataremos neste trabalho de uma ferramenta computacional, que nos permitirá realizar

13 S*i*stema que será desenvolvido como trabalho final da dissertação de mestrado.

análises mais minuciosas sobre um determinado *corpus,* podendo chegar até a uma comparação entre diferentes *corpora*.

Após esta breve definição da abordagem metodológica sobre teoria linguística, com algumas das principais teorias da LC, valemo-nos de alguns desses conceitos para o desenvolvimento deste trabalho. Adiante mostraremos como a LC pode interagir com outras áreas do conhecimento. Neste caso, daremos atenção para a forma como a Ciência da Computação pode se beneficiar com os conceitos elaborados pela LC.

2.5 PRINCÍPIOS DA LINGUÍSTICA DE *CORPUS* - LC

A Linguística é a área do conhecimento em que se desenvolve o estudo científico da linguagem humana com base em fatos linguísticos (MARTINET, 1978). De acordo com Widdowson (1996), de modo geral, os fatos linguísticos podem ser inferidos por meio da introspecção, da licitação e da observação de dados provenientes do uso real da língua pelos seus usuários.

A LC não trata somente da coleta e exploração de *corpora*, ela se ocupa também da descrição das línguas naturais, de aspectos sociais do uso da linguagem, e dos estudos que se dedicam a verificar como a língua vem sendo utilizada por determinadas sociedades. Nesse sentido, por observar as mudanças que ocorreram e que ocorrem na língua, a LC pode ser considerada uma área do conhecimento que sempre está em constante evolução, uma vez que ela parte da observação de *corpora* compostos por dados linguísticos reais.

Outras áreas do conhecimento têm se valido dos preceitos da LC para evoluírem em suas áreas de estudo. A Tradução e o Ensino/aprendizagem de línguas materna e estrangeira, por exemplo, utilizam, cada vez mais, recursos e ferramentas advindos da LC para o ensino de língua estrangeira.

2.6 FERRAMENTAS PARA MANIPULAÇÃO DE *CORPUS*

A LC tem uma relação muito íntima com a tecnologia, uma vez que ela não só nos proporciona meios com os quais podemos obter os dados, mas também para manipulação e armazenamento de forma organizada e estruturada desses dados. A evolução da área está ligada à evolução da computação e à disponibilização de ferramentas para análise de *corpus*.

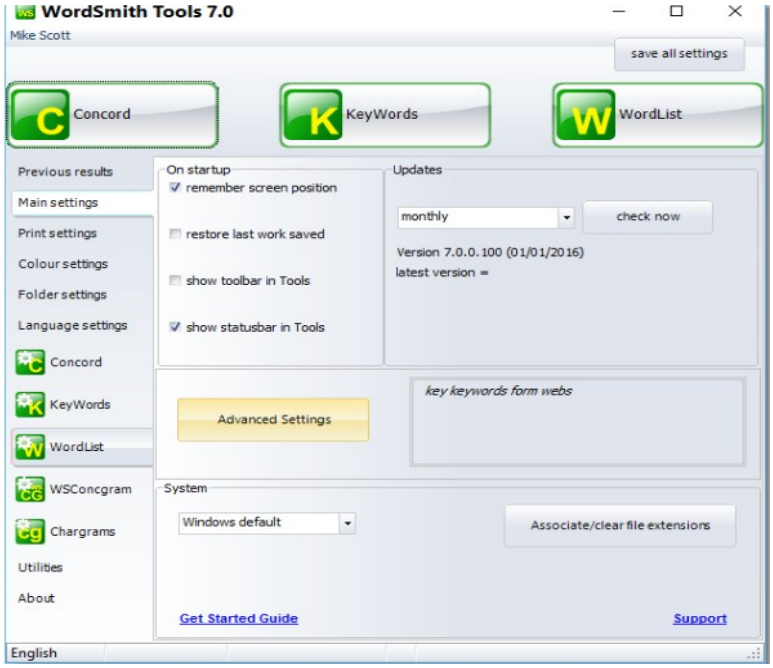
A LC tem se tornado cada vez mais dependente das tecnologias computacionais; por sua vez, as análises linguísticas vão ficando dependentes das evoluções da computação, tendo em vista que é praticamente impossível analisar grandes quantidades de textos sem o uso de algumas das ferramentas computacionais que apresentaremos a seguir.

Abaixo, listamos alguns dos principais programas computacionais mais utilizados em pesquisas que utilizam *corpus*.

2.6.1 *Wordsmith Tools*

O *Wordsmith Tools*, escrito por Mike Scott e publicado pela Universidade de *Oxford University*, é um *software* que disponibiliza uma série de recursos e possui, como principais, as seguintes funcionalidades: *Concord* faz concordâncias; *KeyWords* encontra as palavras-chave em textos; e *WordList* gera lista de palavras em um texto ou em um conjunto de textos. *Site* do programa: http://www.lexically.net/wordsmith.

Figura 9 Tela inicial do programa wordsmith tools



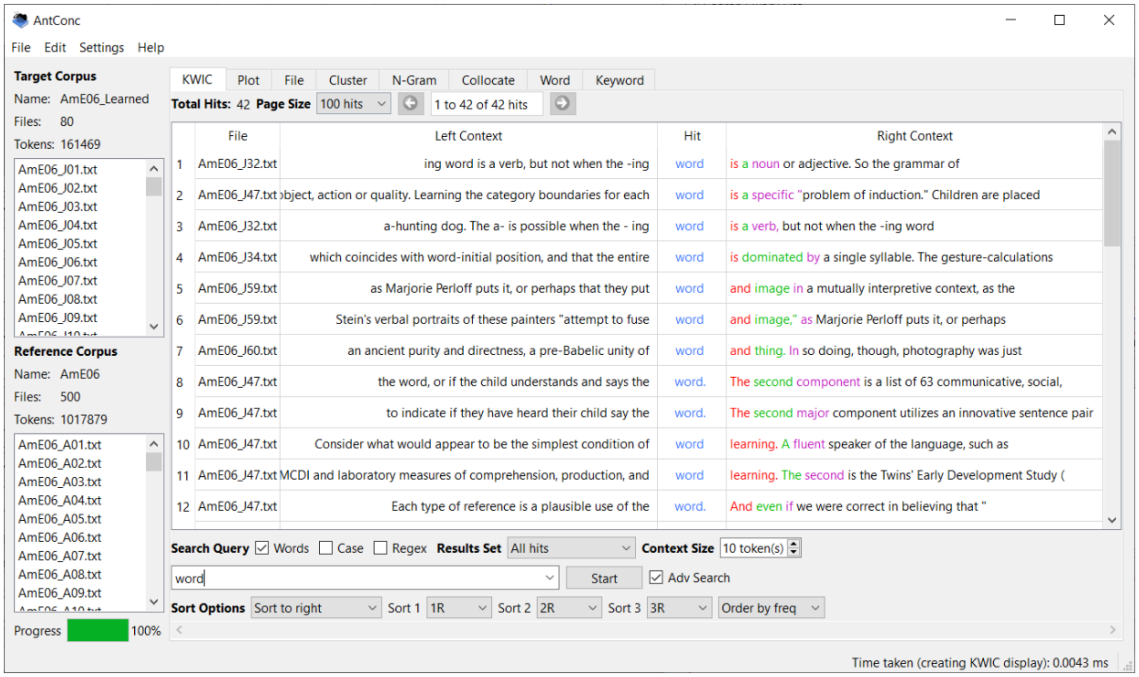
Fonte: https://lexically.net/wordsmith/step_by_step_English7/introduction.html

2.6.2 *AntConc*

O *AntConc* foi desenvolvido por Laurence Anthony. Esse programa está disponível para *download* diretamente do *site* na *internet*. Ele é um utilitário que fornece

uma série de ferramentas para análise e concordância de textos. (Tradução nossa) Site do programa: https://www.laurenceanthony.net/software/antconc/

Figura 10 Tela do programa AntConc

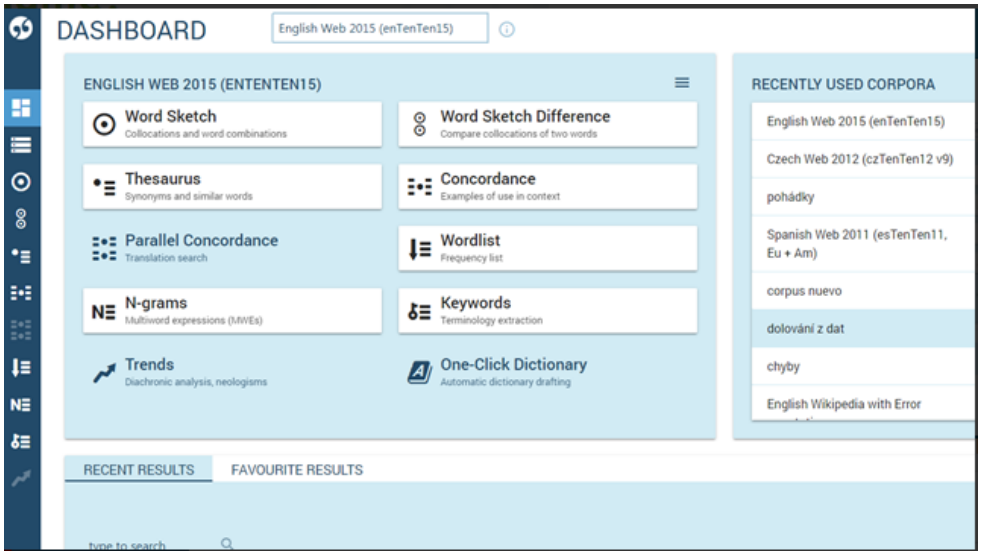


Fonte: https://www.laurenceanthony.net/software/antconc/

2.6.3 *Sketch Engine*

O *Sketch Engine* é uma ferramenta on-line utilizada para gerenciamento de *corpora*. Esse programa é pago, mas pode ser utilizado gratuitamente por um período de 30 dias. Nele, tem-se acesso a todas as funções e ferramentas, a pelo menos um *corpus* por idioma, e a ferramentas de construção de *corpus*. Além disso, o *Sketch Engine* disponibiliza espaço de armazenamento para construir *corpora* de usuários de até um milhão de palavras. Site do programa: https://www.sketchengine.eu/

Figura 11 Dashboard do software sketch engine.

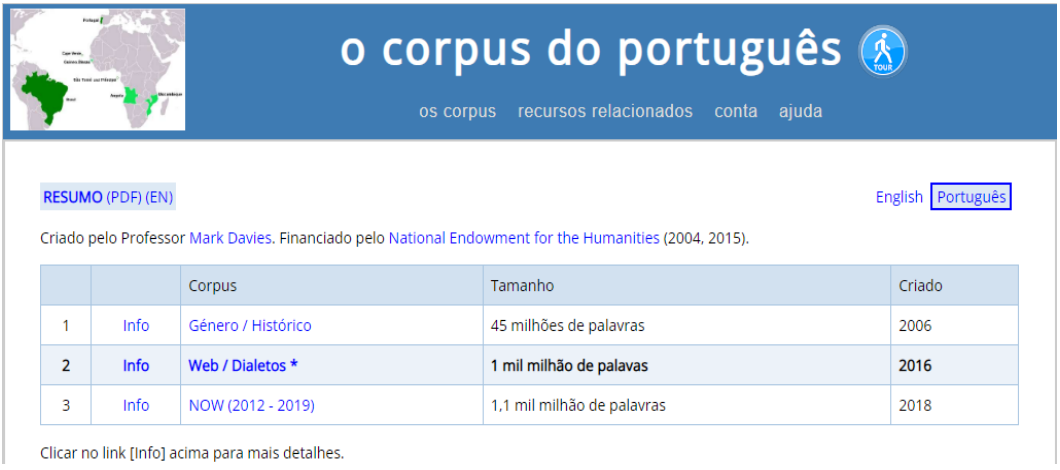


Fonte: https://www.sketchengine.eu/quick-start-guide/

2.6.4 *Corpus do Português*

O *Corpus* do Português, criado por Mark Davies, disponibiliza ferramentas para pesquisa com mais de um bilhão de palavras. Essa ferramenta permite analisar os *corpora* que estão cadastrados em suas bases, mas não possibilita que utilizemos nosso próprio *corpus* para análise. Site do programa: https://www.corpusdoportugues.org/xp.asp

Figura 12 Página inicial do corpus do português



Fonte: https://www.corpusdoportugues.org/xp.asp

2.6.5 *KWIC Key Word in Context*

Key Words in Context (KWIC) refere-se a ocorrências de um elemento em particular nos *corpora* acompanhadas do contexto linguístico em que ele foi empregado (BAKER, 2010, p. 109). Os elementos de uma lista de palavras, utilizados como

referência para a criação de linhas de concordância ou a recorrência dessas palavras, podem levar à descoberta de padrões linguísticos ou à identificação de termos de uma determinada área do conhecimento. A obtenção de listas de palavras, ordenadas de acordo com a frequência em que elas aparecem nos *corpora,* pode ser útil para algumas áreas do conhecimento.

2.7 PROJETO *CLIC DICKENS*

Em pesquisas realizadas na *internet* com o intuito de descobrir bibliotecas e mecanismos que pudessem auxiliar no processamento de textos, os quais são essenciais para esse tipo de trabalho, encontramos a ferramenta *CLiC*, que dispõe de funcionalidades parecidas com as que desejamos implementar em nosso sistema.

O projeto *CLiC Dickens* demonstra, por meio de estilística de *corpus,* como os métodos assistidos por computador podem ser usados para estudar textos literários e levar a novos *insights*[14] sobre como os leitores percebem os personagens fictícios das obras literárias. No âmbito do projeto, foi desenvolvida a aplicação *web CLiC*, desenhada especificamente para a análise de textos literários, mais especificamente sobre as obras de Charles Dickens, um romancista inglês da era vitoriana. *CLiC Dickens* começou a ser desenvolvido na *University of Nottingham,* em 2013, mas agora é um projeto realizado de forma colaborativa, com *a University of Birmingham*.

O projeto em questão está sob a licença *MIT License*[15], conhecida também como Licença X ou Licença X11. Essa é uma licença utilizada em programas de computadores (*software*), criada pelo Instituto de Tecnologia de Massachusetts (MIT), muito utilizada por *softwares* livres. São licenças que concedem um certo grau de liberdade para o usuário, os *softwares* sobe essa licença podem ser executados, ter o seu o código fonte modificado e ser redistribuídos com ou sem modificações.

14 "Insights" é uma palavra em inglês que pode ser traduzida para o português como "percepções", "visões", "entendimentos" ou "ideias". É frequentemente utilizada para se referir a uma compreensão ou entendimento mais profundo sobre um assunto ou situação, geralmente obtido através de observação, análise ou experiência.

15 A Licença MIT (MIT License, em inglês) é uma licença de *software* de código aberto (open source) que permite aos usuários utilizarem, copiarem, modificarem, mesclarem, publicarem, distribuírem, sublicenciarem e/ou venderem o *software*, tanto na forma original quanto em forma modificada, sem restrições significativas. Essa licença é considerada uma das mais permissivas entre as licenças de código aberto, uma vez que é relativamente simples e não impõe muitas restrições sobre o uso do *software*.

A licença MIT é conhecida e muito utilizada por apresentar características mais permissivas. Os produtos licenciados por ela podem ser utilizados de forma mais irrestrita, ou seja, com essa licença é permitido que qualquer pessoa que obtenham uma cópia do *software* e de seus arquivos de documentação possa lidar com eles sem restrições, sem limitação aos direitos de uso, como cópia, modificação, mesclagem, publicação, distribuição e até mesmo a venda de cópias do *software*. Outra vantagem da licença MIT é uma maior clareza ao declarar explicitamente o que é permitido e quais os direitos que estão sendo transferidos com a sua utilização.

Uma questão muito importante diz respeito ao sublicenciamento, em que o *software* é usado como parte integrante de um outro *software*, como quando uma biblioteca adiciona funcionalidades ao *software*. Caso não haja essa liberdade de sublicenciamento, então, apenas o detentor do direito autoral pode conceder a licença. Para utilizar um trabalho, é necessário que o usuário obtenha licença tanto do autor do mesmo quanto dos detentores dos direitos de cada componente que compõem. Para isso, é fundamental identificar todos os componentes e pessoas envolvidas, solicitando o direito de uso correspondente. A única restrição é manter um aviso de *copyright*[16] e uma cópia da licença em todas as cópias do *software*.

Como utilizamos como fonte de análise e de testes um *corpus* da língua portuguesa foi necessário realizar algumas modificações e adequações no sistema para que ele pudesse nos atender, visto que esse sistema foi projetado e desenvolvido para a utilização com textos da Língua Inglesa. Sendo assim, foi necessário fazer a tradução do *software,* dos itens de menu, das mensagens, dos alertas, dentre outros. Outra mudança necessária foi com relação à tratativa que o sistema dá ao texto. No nosso idioma, utilizamos acentuação em algumas palavras para conferir à pronúncia características fonéticas para distingui-la de outras palavras. Essas mudanças não precisam ser realizadas na Língua Inglesa, pois o sistema inicial foi desenvolvido para essa língua.

Quando trabalhamos com análise de texto, temos que nos atentar para alguns detalhes. Por exemplo, algumas palavras, como os conectivos, devem ser filtradas do texto, pois são comumente utilizadas em nossa língua e podem influenciar na contagem de palavras. Conectivos, também conhecidos como conectores ou articuladores do discurso, são palavras ou expressões que ligam frases e orações, permitindo a construção de uma sequência de ideias. A esses conectivos, na área da Ciência da computação, dá-se

16 Direto autoral, ou direito do autor, é um conjunto de prerrogativas conferidas por lei às pessoas físicas ou jurídicas criadoras da obra intelectual.

o nome de *stop word.*[17] A remoção das *stop words* é algo importante, pois elas podem influenciar na contagem de palavras, visto que podem aparecer inúmeras vezes no texto. Convém esclarecer que essa remoção não causa nenhuma perda para o *corpus* em si.

17 *Stop word* são palavras que são comumente usadas em um determinado idioma, mas geralmente não têm significado por si só e, portanto, são frequentemente removidas durante o processamento de linguagem natural (NLP) para melhorar a eficiência e a precisão de algoritmos de análise de texto

CORPUS E METODOLOGIA

Neste capítulo, apresentaremos os *corpora* que serviram de inspiração para o desenvolvimento deste trabalho, eles foram utilizados como base para o desenvolvimento e testes. Esses *corpora* foram compilados por pesquisadores membros do *Grupo em Estudos Contrastivos – GECon*. Estudantes que realizaram ou ainda realizam as suas pesquisas acadêmicas na área da Linguística de Corpus, foram eles os responsáveis por compilar esses *corpora* e disponibilizá-lo para que pudéssemos, com base neles, desenvolver este trabalho. Além da especificidade de cada um desses *corpora*, será apresentada sua extensão, mensurada em número de textos e de itens lexicais.

Após a descrição dos diferentes *corpora* contemplados por esta pesquisa, apresentaremos os diferentes procedimentos metodológicos desenvolvidos ao longo do trabalho. Nesse sentido, faremos a enumeração topicalizada das etapas necessárias para alcançar os objetivos que traçamos para conclusão deste trabalho.

Buscaremos apresentar neste capítulo informações quantitativas e qualitativas a respeito desses *corpora.*

3.1 DESCRIÇÃO DOS CORPORA

Para o presente trabalho, fizemos uso de três *corpora* distintos, os quais foram compilados por outros pesquisadores da área, a saber: Léxico Sertanista; Léxico Indianista; Léxico da Tabatinga; Léxico Toponímico de Goiás em libras; e o Léxico Machadiano. Nos tópicos que se seguem, daremos mais detalhes sobre cada um desses *corpora*.

3.1.1 Léxico Sertanista

O *corpus* intitulado como Léxico Sertanista foi compilado por (Pimenta 2019) como parte do seu trabalho acadêmico para obtenção do título de doutorado. Ele foi utilizado como base para a criação do módulo intitulado com o mesmo nome (Léxico Sertanista). “A autora desse trabalho procurou compilar as obras mais representativas de cada fase do regionalismo, circunscritas em um universo de discurso etnoliterário de

temática sertanista, e que tivessem um repertório lexical diversificado e amplo" (PIMENTA 2019, p.107). Buscando analisar os vocábulos-termos presentes em obras regionalistas brasileiras de temática sertanista, esse trabalho propôs um modelo on-line de representação de *corpus*.

Esse é um dos módulos que compõem a ferramenta *GEConWeb*. Por meio dela o leitor poderá navegar pelas obras que compõem o *corpus* e conhecer alguns dos verbetes que foram identificados pela pesquisadora, além de permitir que o leitor tenha acesso a cada uma das obras e, por meio delas, possa conhecer como cada um desses vocábulos foi empregado no texto.

Para a construção desse *corpus* foram utilizadas as obras regionalistas da literatura Brasileira de autores consagrados não tendo somente o texto como seu objeto primário, mas buscando também um contexto sociocultural, para valorizar a cultura do sertanista.

O *corpus* principal foi dividido em três partes: **RP** (Regionalismo Pitoresco); **RC** (Regionalismo Crítico); e **SR** (Super Regionalismo). Regionalismo Pitoresco: busca enaltecer o homem do sertão e a terra em que vive; Regionalismo Crítico: denuncia os problemas econômicos e sociais das regiões interioranas do Brasil; e Super Regionalismo: é a transcendência do regional para o universal por meio da exploração dos grandes problemas que envolvem o ser humano em qualquer lugar.

Na Tabela 1, trazemos os dados quantitativos do *corpus* Léxico Sertanista e de cada subcorpus[18] que o compõe. Chamamos de subcorpus partes menores que compõem um *corpus*, esses subcorpora normalmente são chamados de subcategorias do *corpus* original.

Tabela 1 - Dados estatísticos do *corpus* Léxico Sertanista

Corpus	***Tokens* (itens)**	***Types* (formas)**	**Razão forma/item %**	**Nº textos**
Corpus geral	1.348.165	71.442	5	20
Subcorpus 1 (RP)	271.201	26.921	10	5
Subcorpus 2 (RC)	568.178	47.422	8	10
Subcorpus 3 (SR)	508.786	33.667	7	5

Fonte: Pimenta (2019, p. 109)

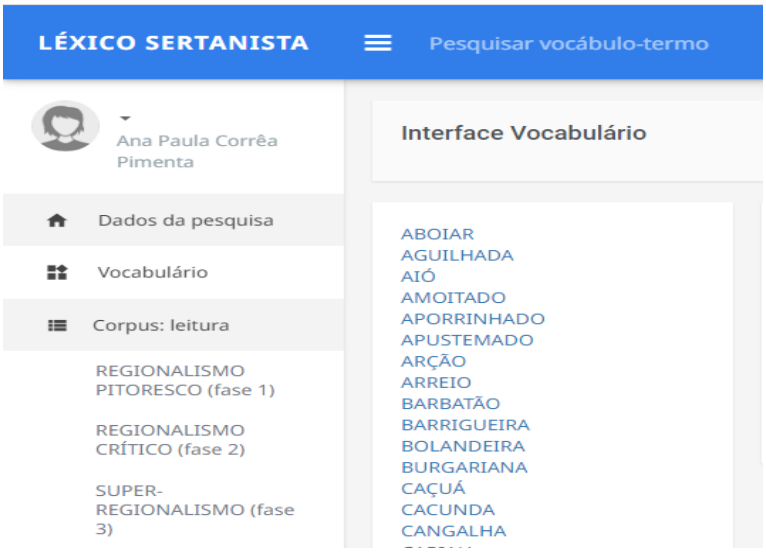
Segundo Pimenta 2019, esses foram os motivos que a fizeram considerar para a escolha dessas obras como representativas de cada uma das fases do regionalismo:

18 Cada *corpus* pode ser dividido em partes menores chamadas subcorpora. Subcorpora pode ser usado para dividir o *corpus* pelo tipo de texto (jornal de ficção), mídia (falada, escrita): Tradução nossa. Disponível em: https://www.sketchengine.eu/guide/create-a-subcorpus/. Acesso: 20 nov. 2022.

> Escolhemos essas obras regionalistas como corpus de análise, em primeiro lugar, por focalizarem o universo sertanejo em seus diferentes aspectos, permitindo-nos conhecer o homem do sertão, suas características físicas, comportamentais e, especialmente linguísticas. Em segundo lugar, por estarem no escopo dos discursos etnoliterários lato sensu – aqueles que utilizam elementos provenientes de discursos etnoliterários stricto sensu (como elementos da literatura oral, do folclore, das lendas e mitos de uma dada cultura), isto é, os discursos-ocorrência do corpus de análise por utilizarem, com frequência, elementos da cultura sertaneja, estabelecendo relações intertextuais e interdiscursivas com o universo de discurso etnoliterário, não apenas se tornam objeto de pesquisa legítimo em Etnoterminologia, como apresentam um material linguístico suficiente e adequado para esta pesquisa (PIMENTA, 2019, p.22).

A figura abaixo ilustra a página inicial do Léxico Sertanista.

Figura 13 Página inicial do Léxico Sertanista



Fonte: https://www.ileel.ufu.br/lexicoSertanista/home

A seguir, descrevemos em mais detalhes o Léxico Indianista, *corpus* compilado por Ávila, 2018.

3.1.2 Léxico Indianista

O *corpus* intitulado como Léxico Indianista foi compilado por Ávila, 2018 como parte do seu trabalho acadêmico para obtenção do título de doutorado. Foram utilizadas as três obras indianistas do autor José de Alencar "Iracema", "O Guarani" e "Ubirajara" - para compilar o *corpus*. As obras desse autor atiçaram na pesquisadora uma curiosidade a respeito de alguns termos utilizados, mesmo ele tendo o cuidado de explicar algumas destas expressões, as que ele julgou serem necessárias, por meio de notas de rodapé. Essa

curiosidade ainda permaneceu com a pesquisadora. Segundo a autora, "O interesse pelos estudos sobre Alencar e sobre seu léxico fora reforçado pelo fato de que, como leitora, sentia a necessidade de conhecer o significado de muitas palavras que o autor utiliza" (Ávila 2018, p. 21). Como exemplo temos abaixo um trecho de texto utilizado pelo escritor com algumas expressões que ele costuma utilizar:

> O nome de Irapuã voa mais longe que o goaná do lago, quando sente a chuva além das serras;
> O búzio dos pescadores do Trairi e a trombeta dos caçadores do Soipé responderam.
> Todos os pescadores em suas jangadas seguiam o chefe e atroavam os ares com o canto de saudade, e os múrmuros do uraçá, que imita os soluços do vento. (ALENCAR, 1965, p. 35).

Nascia aí o fator motivacional que levou a pesquisadora a escolher o tema da sua pesquisa, tendo o seu trabalho inspirado na lexicografia das obras de José de Alencar.

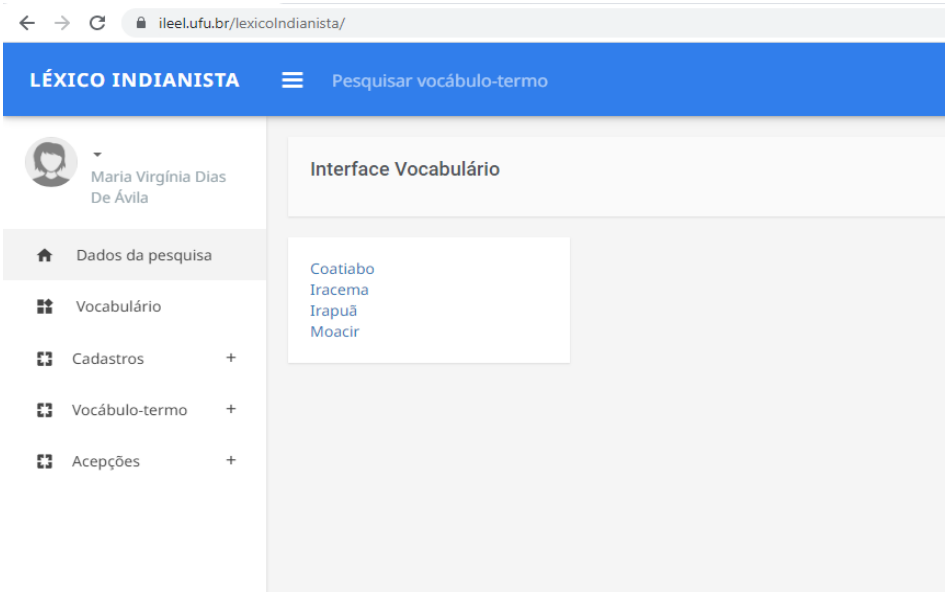
Na Tabela 2, apresentaremos alguns dados estatísticos do *corpus* intitulado como Léxico Indianista, esses dados mostram de uma forma geral como o mesmo foi constituído.

Tabela 2 - Dados estatísticos do *corpus* Léxico Indianista

Corpus	*Tokens* (itens)	*Types* (formas)	Razão forma/item %	Nº textos
Corpus EST[19]	150.524	14.766	10	3

Fonte: Ávila (2018, p. 98)

Figura 14 Página inicial do Léxico Indianista



Fonte: https://www.ileel.ufu.br/lexicoIndianista/

19 EST: Abreviação para Corpus de Estudo.

A seguir descreveremos o Léxico da Tabatinga, trabalho produzido pela pesquisadora Roberta Gê-Acaiaba.

3.1.3 Léxico da Tabatinga

O *corpus* Léxico da Tabatinga encontra-se em fase de compilação pela pesquisadora (Gê-Acaiaba, 2023) como parte da sua pesquisa de doutorado. Esse trabalho de pesquisa buscou resgatar uma língua que pouco se conhece e que falam, a língua da Tabatinga[20]. Língua dos Negros da Tabatinga da região de Bom Despacho, cidade que fica a aproximadamente 160km da capital mineira, Belo Horizonte.

Esse é um estudo que trata da descrição e análise da Língua da Tabatinga a partir da Linguística de Corpus. A Língua da Tabatinga se formou da fusão do português brasileiro com línguas de origem banto.

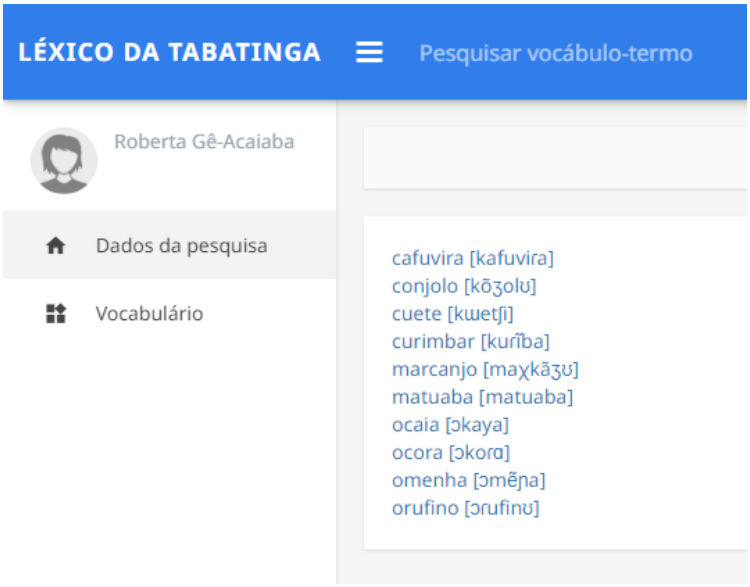
O objetivo do léxico da língua de Tabatinga é evidenciar que não existem comunidades linguísticas homogêneas e que a heterogeneidade na formação ou evolução de uma língua é um fenômeno comum e natural. Segundo a pesquisadora: "a dinâmica das línguas, organismos vivos que se moldam ao mesmo tempo em que moldam a sociedade na qual se desenvolvem". (GÊ-ACAIABA, 2022, p.7).

Partindo do pressuposto de que a língua é um ser em constante evolução, surgiu assim o interesse por parte da pesquisadora em investigar o processo de formação da Língua da Tabatinga, e verificar como ela interfere na constituição da Língua Portuguesa Brasileira falada na cidade de Bom Despacho (GÊ-ACAIABA, 2022, p.7).

A seguir temos uma figura ilustrando a página inicial do Léxico da Tabatinga.

20 Disponível em: https://pingodeouvido.com/tag/lingua-da-tabatinga-em-bom-despacho/. Acesso em: 17 nov. 2022.

Figura 15 Página inicial do Léxico da Tabatinga



Fonte: https://www.ileel.ufu.br/lexicoTabatinga/

A seguir, apresentaremos o Léxico Toponímico de Goiás em Libras, trabalho produzido pela pesquisadora Mariano 2023.

3.1.4 Léxico Toponímico de Goiás em Libras

O Léxico Toponímico de Goiás em Libras é um módulo um pouco diferente dos demais, com algumas peculiaridades. Uma característica em particular desse *corpus* está justamente na Libras[21], sigla utilizada para se referir à Língua Brasileira de Sinais, uma língua gestual-visual que permite a comunicação por meio de gestos, expressões faciais e corporais.

Sabendo que, por ser a Libras uma língua como uma outra qualquer, podemos então entender que os métodos e ferramentas disponíveis aliados à LC podem também ser utilizados em pesquisas linguísticas em Línguas de Sinais. Berber Sardinha (2004) pontua que a LC vem mudando o modo de investigar a linguagem, tendo em vista que por meio dela o pesquisador tem acesso a conjuntos de textos e transcrições de falas em quantidade antes inacessíveis, E que os estudos linguísticos têm a cada dia tomado grande proporção nas pesquisas brasileiras. Ferreira Brito (1995 p. 29) observa que: “As pesquisas sobre as línguas de sinais têm demonstrado quão complexa, completa, abstrata

21 Disponível em: https://www.libras.com.br/#:~:text=O%20que%20%C3%A9%20Libras%3F,-O%20QUE%20%C3%89&text=Libras%20%C3%A9%20a%20sigla%20da,atrav%C3%A9s%20da%20Lei%20n%C2%BA%2010.436. Acessado em: 17 nov. 2022.

e rica pode ser uma modalidade visual de língua".

O léxico, como parte integrante de qualquer língua, adapta-se conforme a necessidade do ser humano, evolui junto com ele. Nomear e categorizar seres, objetos e espaços são algumas das aplicações nas quais ele é utilizado. Barbosa (1992, p. 122) nos lembra que "todo sistema linguístico contém unidades lexicais, inventário à disposição dos falantes, unidades estruturadas de acordo com regras que permitem aos usuários a criação de novas palavras mais adequadas as suas necessidades de comunicação".

No trabalho realizado pela pesquisadora Kássia Mariano, estabeleceu-se uma interface entre os estudos toponímicos e a Libras, possibilitando o registro, a descrição e a análise dos signos toponímicos em Libras. O léxico é considerado uma parte essencial de uma língua, pois é responsável por nomear espaços e refletir os aspectos culturais de um povo. É através dessa relação que ele consegue estabelecer sua importância. Segundo Mariano (2022, p. 8):

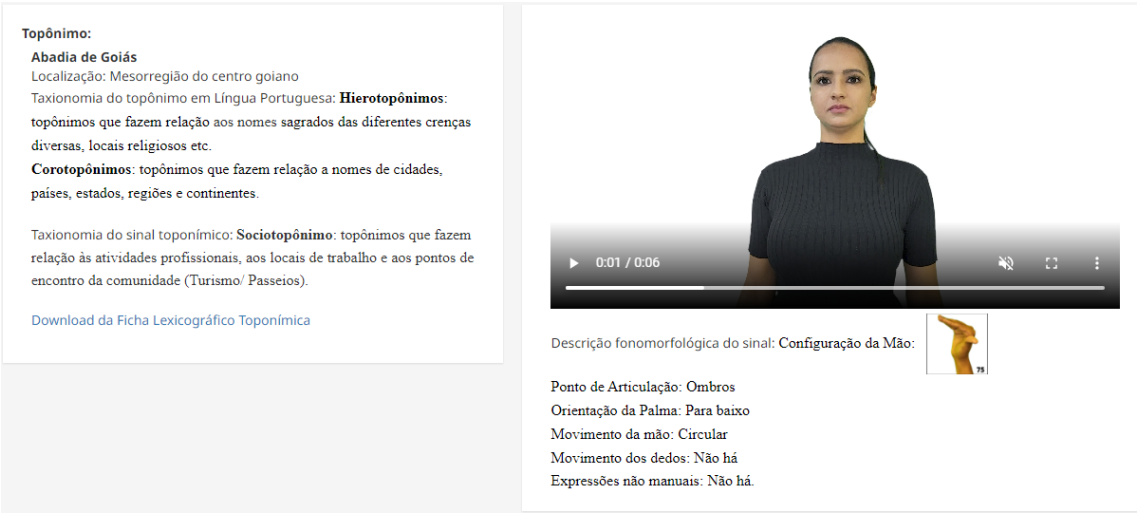
> Dentro da Toponímia o signo linguístico é representado por topônimos, isto é, nomes próprios que designam um espaço geográfico. Sendo a Libras um sistema linguístico organizado, deve ser capaz de, assim como qualquer outra língua, nomear os espaços geográficos por meio de sinais, que são as unidades formadoras do conjunto lexical da língua.

Ainda segundo a autora, "Foi a partir dessa assertiva, que foram direcionados os esforços para à apreensão, registro, descrição e análise motivacional dos sinais toponímicos de cidades do Estado de Goiás, a fim de averiguar de que maneira se realiza em Libras a nomeação das cidades e municípios goianos" (MARIANO, 2022 p.16). Foi, portanto, esse o fator motivacional que a pesquisadora encontrou para a realização do seu trabalho.

Foram traduzidos para a língua de sinais os topônimos das cidades de Goiás. A partir desse trabalho que se encontra em fase de desenvolvimento é possível listar alguns dos nomes das cidades do estado de Goiás com a sua respectiva tradução em Libras.

Abaixo, temos uma figura ilustrando como será a pesquisa por topônimos no nesse módulo.

Figura 16 Léxico Toponímico em Libras



Fonte: https://www.ileel.ufu.br/topominiaLibras/

A seguir, descreveremos o Léxico Machadiano, *corpus* produzido pelo autor deste trabalho.

3.1.5 Léxico Machadiano

Com o intuito de validar e testar a viabilidade da ideia, utilizamos as obras do escritor brasileiro Machado de Assis como fonte de teste e de análise para esta pesquisa. As obras do escritor foram escolhidas em razão da sua relevância para a literatura brasileira. Outro ponto que veio corroborar com a utilização dessas obras foi o fato delas se encontrarem em domínio público, ou seja, por não incidirem sobre elas direitos autorais, podendo, portanto, ser reproduzidas por qualquer pessoa sem que haja alguma infração sobre direitos autorais.

O Ministério da Educação (MEC) disponibilizou, por meio do seguinte endereço eletrônico: http://machado.mec.gov.br/index.php, o acesso a todas as obras do autor. Além da obra completa, esse portal nos permite acessar outras informações como, por exemplo: cronologia, bibliografia, vídeos. A **Figura** *17*, mostra a página inicial do portal disponibilizado pelo MEC com as obras do autor.

Figura 17 Portal com as obras do autor

Fonte: http://machado.mec.gov.br/index.php

O *corpus* intitulado como Léxico Machadiano foi compilado por este pesquisador a partir das obras de Machado de Assis. Essas obras foram coletadas a partir do Portal Domínio Público, onde se encontram todas as obras desse escritor brasileiro disponíveis para acesso. Na Tabela 3 apresentamos os dados estatísticos do *corpus* por subcategoria, que são elas: RM (Romance); CT (Contos); PE (Poesia); CR (Crônica); TE (Teatro); CR (Crítica); TR (Tradução); e MI (Miscelânea).

Tabela 3 - Dados estatísticos do *corpus* Léxico Machadiano

Corpus	*Tokens* (itens)	*Types* (formas)	Razão forma/item %	Nº textos
Corpus geral	3.121.944	79.461	3	246
Subcorpus 1 (RM)[22]	662.346	31.493	21,3	10
Subcorpus 2 (CT)	1.143.580	39.906	28,6	137
Subcorpus 3 (PE)	144.389	19.418	7,4	7
Subcorpus 4 (CRI)	195.828	17.839	10,9	45
Subcorpus 5 (TE)	83.558	9.311	8,9	10
Subcorpus 6 (CR)	195.828	40.920	5	24
Subcorpus 7 (TR)	227.835	20.786	11	3
Subcorpus 8 (MI)	14.415	3.813	4	10

Fonte: O autor

A seguir, descrevemos os procedimentos pertinentes à metodologia aplicada.

22 RM (Romance); CT (Contos); PE (Poesia); CR (Crônica); TE (Teatro); CRI (Crítica); TR (Tradução); MI (Miscelânea).

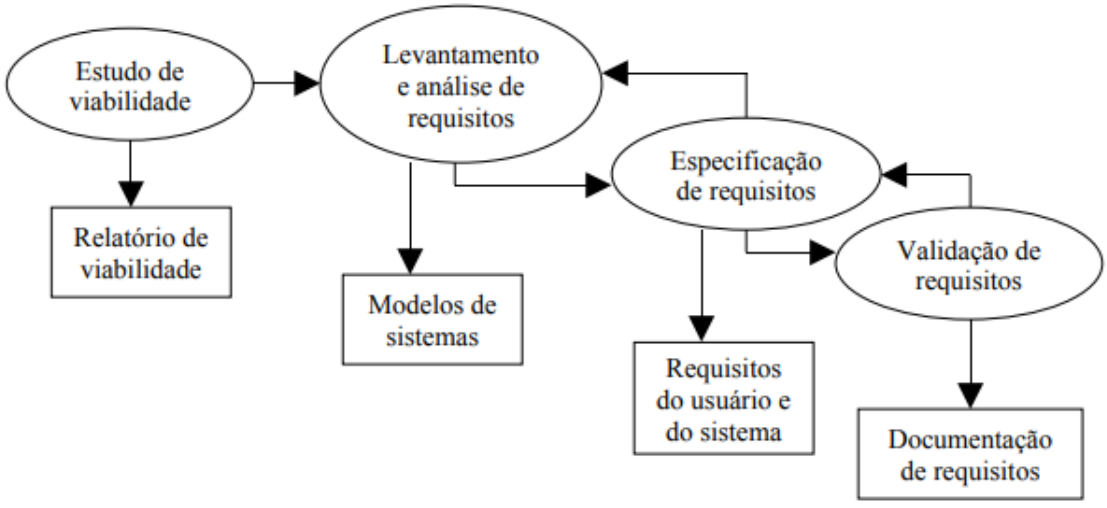
3.2 DESCRIÇÃO DOS PROCEDIMENTOS METODOLÓGICOS

Após diversas reuniões realizadas com as diferentes pesquisadoras no intuito de compreender as necessidades referentes às respectivas pesquisas, passamos à modelagem da plataforma. O intuito dessas reuniões foi entender e compreender quais eram as necessidades e os anseios das pesquisadoras quanto ao produto de *software* a ser desenvolvido.

Os levantamentos necessários para definir as funções e restrições de um produto de *software* são chamados de requisitos de *software,* que se constituem como uma descrição das funções e restrições que o produto de *software* deve atender. De acordo com Summervile (2003, p.99), "os requisitos para um sistema de software estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e implementação". Deste modo, a análise de requisitos é a etapa do desenvolvimento de um *software*, na qual a equipe de desenvolvimento lista quais são as necessidades do usuário, o contexto no qual o *software* será inserido, além das funcionalidades que serão criadas. A partir dessa análise é que é gerada a modelagem do *software*.

Veja na imagem abaixo os procedimentos que são adotados durante a fase de análise de requisitos.

Figura 18 Modelagem da análise de requisitos



Fonte: Pressman, 2003

Na tabela abaixo damos mais detalhes de cada uma dessas fases, especificando quais dados cada uma delas é responsável por coletar e qual informação ela produzirá para o processo.

Quadro 4 - Descrição do processo de análise de requisitos

Fases do processo	Descrição da fase
Estudo de viabilidade	É um estudo de viabilidade, é uma análise técnica e econômica, utilizado para verificar se as necessidades que foram levantadas podem ser satisfeitas com a utilização das atuais tecnologias de *hardware* e *software*. Este estudo decidirá se o sistema proposto é viável financeiramente ou não.
Levantamento e análise de requisitos	É um processo que visa identificar, coletar, analisar e documentar as necessidades e expectativas dos usuários de um sistema de *software* ou de uma solução tecnológica.
Especificação de requisitos	A especificação de requisitos é o processo de detalhar e documentar os requisitos funcionais e não funcionais de um sistema ou *software* em um documento formal.
Validação de requisitos	Essa atividade verifica os requisitos quanto a sua pertinência, consistência e integralidade. Durante esse processo inevitavelmente são encontrados erros na documentação de requisitos. Os requisitos devem então ser modificados, objetivando corrigir esses problemas.

Fonte: Summerville 2003

Segundo essa metodologia de desenvolvimento de *software*, o processo se torna um conjunto de atividades e procedimentos que associados geram um produto, neste caso é o *software* em si, e essa metodologia possui basicamente quatro atividades fundamentais, a saber (SUMMERVILLE, 2003, p.7):

1. Especificação de *Software*: Definição de funcionamento e restrições.
2. Desenvolvimento de *Software*: Construção baseada nas especificações.
3. Validação do *Software*: Testes que garantam a qualidade do produto.
4. Evolução do *Software*: Melhorias que visam atender novas demandas.

Seguindo os fundamentos defendidos pela engenharia de *software* citados acima, apresentamos a seguir uma estrutura topicalizada, com as etapas que se sucederam durante o desenvolvimento desta pesquisa:

a) Escolha de qual banco de dados utilizar;
b) Modelagem das entidades do banco de dados;
c) Escolha da linguagem de programação;
d) Arquitetura da plataforma;
e) Análise e desenvolvimento de *software*;

3.2.1 Escolha de qual banco de dados utilizar

Para escolher qual Sistema Gerenciado de Banco de Dados - SGBD[23] iríamos utilizar em nossa aplicação levamos em consideração alguns fatores importantes, tais como: Segurança, Integridade dos dados, Controle de concorrência, recuperação e tolerância a falhas.

Existem atualmente diversos SGBD's que poderíamos escolher para ser utilizado no nosso projeto, como: IBM Informix, PostgreSQL, Firebird, HSQLDB, MySQL, Oracle, SQL-Server, dentre outros, qualquer um deles atenderiam aos requisitos do nosso sistema. Esses foram alguns exemplos de SGBD's que poderímos utilizar para a nossa aplicação, porém, como se trata de um trabalho acadêmico sem fins lucrativos e que não foi subsidiado por nenhuma verba, tivemos que renunciar a alguns deles por serem produtos comercializados por empresas como a Microsoft e a Oracle. Essas empresas são as donas dos bancos SQL-Server e Oracle, respectivamente.

Levando-se em consideração que não foi possível utilizar um produto que precisasse da aquisição de uma licença, tivemos então que observar um outro requisito, que o SGBD escolhido atendesse ao requisito mensurado acima e que fosse um *software* livre, que não requer a compra de uma licença. Na lista acima tivemos alguns modelos que atendiam a esse novo requisito, são eles: PostgreSQL, Firebird, HSQLDB e MySQL.

Dentre essa lista, optamos por utilizar o MySQL, a escolha se deu pelo simples fato de ele ser o SGBD que já estava sendo utilizado no servidor onde hospedaríamos o sistema e assim evitamos um trabalho de instalação e configuração do banco de dados no servidor.

Feita a escolha do SGBD com base em critérios técnicos e econômicos, partimos então para a próxima etapa, dando início à modelagem do banco, à criação das tabelas, os campos das tabelas e dos relacionamentos entre essas tabelas. Demonstramos a seguir os procedimentos adotados nessa próxima etapa.

3.2.2 Modelagem das entidades do banco de dados

23 SGBD e *software* utilizado para gerir Bases de Dados, permitindo criar base de dados, modificar Base de dados, eliminar bases de dados, inserir dados na Base de Dados e Eliminar dados da base de dados. Disponível em: https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788#:~:text=Os%20SGBD%20(Sistemas%20de%20Gest%C3%A3o,falhas%20(Backup%20e%20Transactionlogging). Acessado em: 21 nov. 2022.

Uma das fases mais cruciais no processo de desenvolvimento de um sistema de *software* é a criação de um projeto de banco de dados. Para elaboração desse banco devemos seguir algumas etapas, partindo do levantamento de requisitos até a definição das entidades e dos campos que serão armazenados pelo banco de dados.

Após a identificação das entidades e dos campos que cada uma das entidades teria, identificamos quais são as relações que elas manteriam umas com as outras. Abaixo ilustramos uma entidade que foi identificada na fase de análise de requisitos. A entidade em questão se chama **trabalho e** armazena os dados referentes ao trabalho/pesquisa que foi realizado. Essa entidade foi convertida em uma tabela dentro do banco de dados, onde forma registrados os dados referentes ao trabalho.

Figura 19 Entidade trabalho

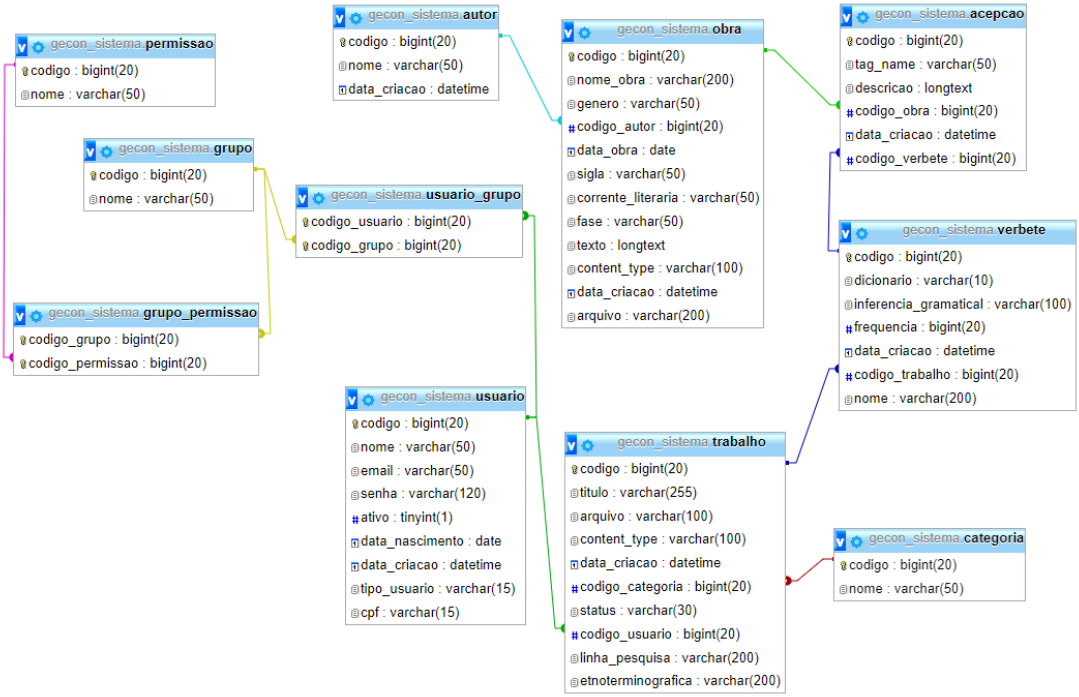
Fonte: O autor

Como produto final da fase de análise de requisitos foi gerado um documento chamado de Diagrama de Entidades e Relacionamento – DER[24]. Na Engenharia de Software, o modelo conceitual conhecido como DER é empregado para representar as entidades envolvidas em um domínio de negócios, juntamente com seus atributos e as relações existentes entre elas.

Na figura abaixo mostramos como ficou o Diagrama de Entidades e Relacionamento – DER produzido na faze anterior e que, representa a estrutura lógica da aplicação e as suas relações, além de representar também a estrutura de armazenamento do banco de dados.

24 Disponível em: https://www.alura.com.br/artigos/mer-e-der-funcoes. Acesso em: 23 nov. 2022.

Figura 20 Diagrama de Entidade e Relacionamento – DER



Fonte: O autor

Depois da fase de análise dos requisitos e geração do DER, partimos para a etapa engloba a escolha de qual linguagem de programação utilizar no desenvolvimento do sistema. No item que se segue damos mais detalhes dos itens que levamos em consideração para a escolha da linguagem.

3.2.3 Escolha da linguagem de programação

Quais itens devemos levar em consideração para a escolha da linguagem mais adequada para o desenvolvimento do seu projeto? Primeiramente, devemos decidir para qual plataforma desejamos desenvolver, pois, há linguagens cujo uso tem um objetivo em específico, já outras linguagens, podem ser utilizadas em várias plataformas.

Se o foco do projeto é ser utilizado por intermédio da *internet*, ou seja, um *site* ou um sistema *web,* devemos levar em consideração linguagens que possibilitem esse tipo de abordagem. No desenvolvimento de sistemas *web*, existe um detalhe que deve ser levado em consideração, os sistemas ou sites *web* podem ser divididos nas seguintes

partes: Front-End[25] é Back-End[26] ou Full-Stack[27].

Front-End refere-se a algo que está na parte frontal, neste caso, estamos nos referindo à frente do sistema, a parte visual, ou seja, a parte na qual o usuário verá e interagirá com o sistema. Back-End é o oposto disso, é a parte do sistema que põe em prática as regras do negócio, ou seja, é responsável por todas as funcionalidades.

De forma bem resumida, o front-end é a parte responsável pelo *layout* da aplicação que o usuário irá ver e interagir, já o back-end é o responsável pelo que é executado no servidor. O full-stack engloba todas essas funcionalidades juntas em uma atividade.

Para o desenvolvimento deste trabalho optamos por uma linguagem que fosse multiplataforma e com um paradigma mais atual. Uma linguagem multiplataforma se faz necessária devido à gama de sistemas operacionais que podemos utilizar, além de nos ajudar no desenvolvimento ela nos possibilita a utilização de qualquer computador para a atividade de desenvolvimento. Quanto à hospedagem do sistema, ela nos dá mais flexibilidade na hora de escolher qual sistema operacional poderemos instalar no servidor para executar o nosso sistema.

Seguindo nessa linha, optamos pela linguagem Java, pois ela atende a todos os requisitos de uma linguagem multiplataforma e com um paradigma atual de construção de sistemas.

3.2.4 Análise e desenvolvimento de *software*

Um programa de computador (*software*[28]) pode ser basicamente definido como um conjunto de instruções sequenciais ou como um conjunto organizado de instruções que foram escritas em uma determinada linguagem de programação, permitindo, assim, que os computadores realizem as mais diversas tarefas. Segundo Pressman (2010, p. 636), um *software* é um conjunto composto por instruções de computador, estruturas de dados e documentos.

---

25 É a parte responsável pelo layout da aplicação, a parte que o usuário vai interagir com o sistema. Fonte: https://metodoprogramar.com.br/como-se-tornar-um-desenvolvedor-front-end/. Acesso em: 23 nov. 2022

26 É a parte responsável pelas regras de negócio da aplicação. Fonte: https://metodoprogramar.com.br/como-se-tornar-um-desenvolvedor-web-back-end/. Acesso em: 23 nov. 2022

27 Método de desenvolvimento que envolve as duas modalidades: front-end e back-end.

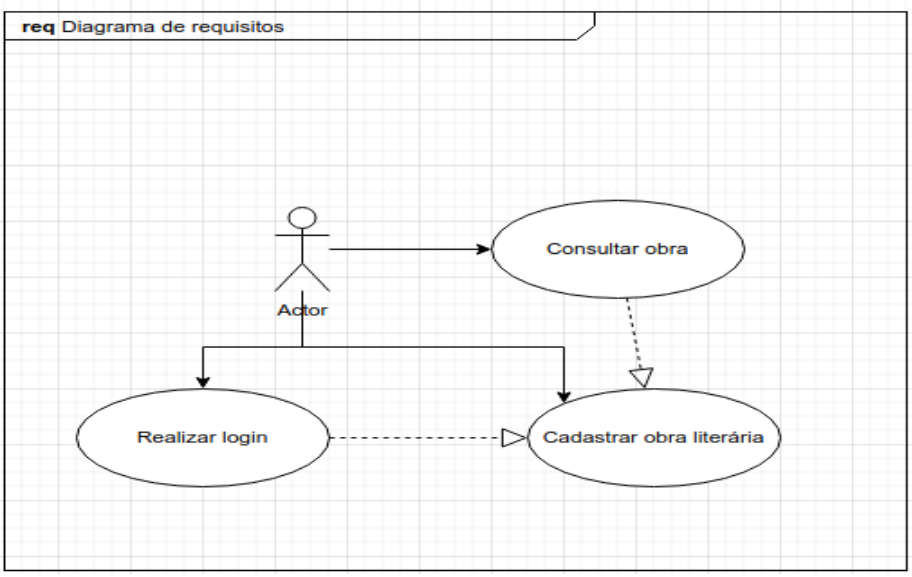
28 O termo inglês *software* foi usado pela primeira vez em 1958 em um artigo escrito pelo cientista americano John Wilder Tukey. Retirado de http://taleron.com.br/2016/10/26/a-evolucao-do-software/ acessado em 14 de junho 2022.

Quando pretendemos desenvolver algum programa de computador, buscamos entender as necessidades que levaram a esse desenvolvimento, para essa tarefa é dado o nome de modelagem de requisitos. Na modelagem de requisitos nos, focamos no objetivo principal, no problema que pretendemos solucionar com o desenvolvimento do *software*, e não em como faremos para resolvê-lo.

A análise de requisitos resulta em uma série de especificações e características operacionais do *software*, tais como: i) Quais ações ele deve executar?; ii) Quais respostas ele deve apresentar?; iii) Qual resposta é esperada, para cada ação do usuário?; iv) Quais objetos o sistema manipulará?; v) Quais funções o sistema deve executar?; vi) Quais comportamentos o sistema apresentará; vii) Quais interfaces são definidas?; e viii) Quais restrições se aplicam?

Apresentamos, a seguir, a **Figura *21***, que ilustra essa atividade de análise de requisitos, produto que é gerado pela fase de análise de requisitos.

Figura 21 Diagrama de caso de uso



Fonte: O autor

A análise das funcionalidades que o *software* terá que desempenhar é também conhecida como análise de domínio ou análise de caso de uso. Firesmith (1993, p.8) descreve a análise de domínio da seguinte maneira:

> Análise de domínio de um *software* é a identificação, a análise e a especificação de requisitos comuns de um campo de aplicação específico, tipicamente para reutilização em vários projetos dentro

> deste campo de aplicação... [Análise de domínio orientada a objetos é] a identificação, análise e especificação de capacidades comuns reutilizáveis dentro de um campo de aplicação específico, em termos de objetos, classes, componentes e frameworks[29] comuns (Tradução nossa).

Após a fase de análise e especificação dos requisitos funcionais do sistema, é dado início à fase de estudo e planejamento da arquitetura.

3.2.5 Arquitetura da Plataforma

Antes de iniciar o desenvolvimento de qualquer projeto de *software,* devemos primeiro iniciar o projeto de arquitetura do mesmo, que vem antes da etapa de construção em si. O projeto de arquitetura define as partes e suas funções, bem como a forma como devem se relacionar entre si. Desse modo, a arquitetura assegura a coerência do *software*, ou seja, a consistência entre suas diversas partes. Segundo Perry e Wolf (1992), *software* é um conjunto de elementos arquiteturais (de dados, de processamento, de conexão) que possui alguma organização. Os elementos e sua organização são definidos por decisões tomadas para satisfazer objetivos e restrições. De acordo com Shaw e Garlan (1996), Arquitetura de Software se refere à descrição dos elementos com os quais um sistema é construído, as interações entre esses elementos, os padrões que guiam a sua composição e as restrições nesses padrões. Em linhas gerais, a arquitetura de um *software* definirá como ele irá se comportar e quais elementos o comporão.

Além da simples escolha de linguagem e banco de dados para o nosso sistema, a arquitetura de *software* desempenha um papel fundamental. Ela não só nos ajuda a selecionar algoritmos e estruturas de dados adequados, mas também nos fornece orientação para responder perguntas importantes como: i) Como será a estruturas do sistema?; ii) Quais controles teremos?; iii) Quais são os protocolos de comunicação, sincronização e acesso a dados?; Como é feita a atribuição de funcionalidades aos elementos do sistema?; Qual é a distribuição física dos componentes: vi), como a arquitetura de *software* abordará a escalabilidade, o desempenho e outros atributos de qualidade?

Esses e outros fatores compreendem o projeto em nível arquitetural e estão diretamente relacionados com a organização do sistema e, portanto, afetam os atributos

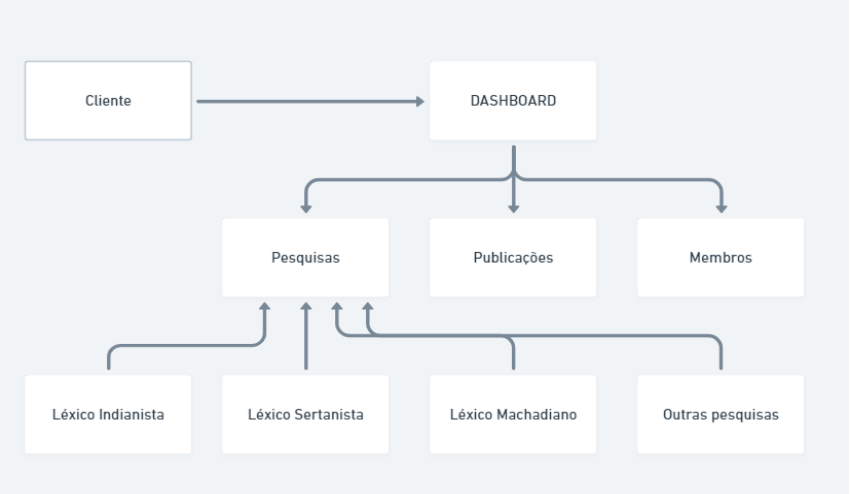
29 São *softwares* que trazem funcionalidades já prontas e que são usados no desenvolvimento de aplicativos e *sites*, evitando, assim, a necessidade criar determinadas tarefas.

de qualidade (também chamados de requisitos não funcionais), como desempenho, portabilidade, confiabilidade, disponibilidade, entre outros.

No desenvolvimento de uma aplicação, são tomadas várias decisões importantes, que são norteadas com base em qual estilo arquitetural será utilizado, tais como: qual a sua construção; qual arquitetura utilizar; se será uma aplicação que será executada na *internet* (*web*) ou se será uma aplicação que será executada em um computador local (*desktop)*. Podemos observar pela **Figura *22*** como foram dispostos o fluxo e a organização das páginas da aplicação.

O componente cliente, representando na **Figura *22***, corresponde ao navegador do usuário, que acessará a página inicial do *site* ou, como comumente chamamos, *dashboard*[30],e, a partir dela, ele poderá navegar por entre os recursos e funcionalidades do *site.*

Figura 22 Estrutura da aplicação



Fonte: o autor

Com auxílio da arquitetura de *software* será possível entender as diferenças entre as linguagens, entre os sistemas operacionais e ambientes computacionais, ou seja, qualquer componente tecnológico poderá ser utilizado para integrar uma solução arquitetural da aplicação. Ela é essencial, pois, é por meio dela que podemos otimizar e organizar o trabalho dos desenvolvedores, permitindo que a aplicação seja desenvolvida dentro do projetado.

Antes de darmos início ao desenvolvimento da ferramenta propriamente dita tivemos algumas etapas importantes para realizar. Alguns procedimentos formam

30 Página inicial ou um painel visual que contém informações iniciais de um sistema ou *site*.

seguidos para que pudéssemos ter um material adequado para testes e validação, alguns desses passos são: baixar os textos, limpar, organizar, formatar e preparar os textos para serem utilizados com um *corpus*.

Descrevemos a seguir, de forma mais detalhada, esses procedimentos e como eles foram criados e executados.

3.2.6 Baixando *corpus* da *internet* com auxílio de *scripts* de programação

O objeto de estudo da Linguística é a língua e, mais especificamente, para a Linguística de *corpus* são os textos, que tenham sido escritos em linguagem. Segundo Biderman (2001, p.79), podemos definir *corpus* como: uma coletânea de textos em formato eletrônico codificados de modo padronizado e homogêneo. Tendo como base esse princípio, podemos afirmar que a principal forma de obtenção desses dados da atualidade se dá por intermédio da *internet*, ou através da digitalização de documentos, tornando-os assim documentos digitais. Os textos disponibilizados via *internet* estão comumente em formato de hipertexto[31] ou em algum outro formato eletrônico.

A captura desses textos pode ser uma tarefa demorada e cansativa. Em se tratando de documentos em formato HTML[32], a coleta pode ser feita basicamente com comandos, de seleção[33], cópia[34] e cola[35]. Para a captura de textos em formato PDF[36], o procedimento anterior não se aplica. Por se tratar de um formato diferente de documento, é necessário outro procedimento. Portanto, para realizar essa tarefa, o pesquisador deve fazer o *download* manual do documento, utilizando as funcionalidades de *download* de arquivos disponibilizadas pelo navegador da *internet*.

Dentre as questões que podem influenciar na criação de um *corpus,* o tempo e o esforço para a elaboração manual do mesmo estão entre as que mais consomem o

31 De acordo com Baker, Hardie e Mcenery (2006), um documento de hipertexto pode conter *links* para outros documentos e formar redes de textos. Os documentos de hipertexto estão presentes na *Internet* sob o formato HTML, uma derivação do Standard Generalised Markup Language (SGML).
32 Linguagem de marcação de hipertexto é um bloco de construção mais básico da *internet*.
33 Para selecionar texto em navegadores de *internet* podemos usar a combinação de teclas CTRL + A.
34 Ao pressionarmos a sequência de teclas CTRL+C simultaneamente realizamos a cópia de um texto que esteja selecionado.
35 Ao pressionarmos a sequência de teclas CTRL+V simultaneamente realizamos o procedimento de colar texto que esteja na área de transferência do sistema operacional.
36 PDF é uma sigla que significa Portable Document Format (Formato de Documento Portátil, em português). É um formato de arquivo desenvolvido pela Adobe Systems que permite a apresentação de documentos em um formato independente de hardware, software e sistema operacional. O objetivo principal do PDF é permitir que documentos sejam visualizados e impressos de maneira idêntica em qualquer dispositivo, mantendo a formatação original do documento.

pesquisador, sendo que sua intervenção ganha destaque ao estar presente em praticamente todas as fases do processo (BAKER, 2010, p. 109). Para tornar essas tarefas menos onerosas, objetivamos criar uma ferramenta que pudesse fazer esse trabalho de forma mais rápida e prática. Para tanto, não só utilizamos todo o poder e a flexibilidade que as linguagens de programação nos proporcionam, mas também desenvolvemos um *script*[37] para fazer o *download* das obras do escritor brasileiro Machado de Assis, que estão disponíveis em domínio público[38] em formato PDF e as convertemos em arquivo de texto simples TXT[39]. Neste processo automatizado de captura dos textos não se fez necessário nenhuma conversão dos textos de PDF para TXT, o *script* se encarregou dessa tarefa para nós.

3.2.7 Procedimentos para criação do *script* de coleta das obras

Conforme dito em capítulos anteriores, utilizamos a linguagem *Python* para a criação do *script* responsável por realizar o *download* das obras do escritor Machado de Assis. Para iniciarmos o procedimento de extração dos textos, em primeiro lugar, acessamos o *site* no qual as obras estão hospedadas (http://machado.mec.gov.br). Logo, na página inicial, há um item de *menu* chamado "Obra completa". Na **Figura *23***, a seguir, apresentamos o item de menu do *site*.

37 Arquivo em formato texto onde são escritas sequências lógicas e procedimentos para executar determinada tarefa.
38 Domínio público é um estado em que as obras intelectuais ou materiais não possuem mais direitos autorais e, portanto, podem ser utilizadas livremente por qualquer pessoa, sem a necessidade de permissão do autor ou pagamento de royalties. O termo "domínio público" se aplica a obras cujos direitos autorais expiraram ou foram renunciados pelo autor. Isso significa que a obra pertence a todos e pode ser copiada, distribuída, traduzida e adaptada sem restrições.
39 TXT é uma sigla que significa Texto (em inglês, Text). Um arquivo TXT é um arquivo de texto simples que contém apenas texto sem formatação, como fontes, cores e outros elementos de *layout*.

Figura 23 Item de menu Obra completa

Fonte: http://machado.mec.gov.br/index.php

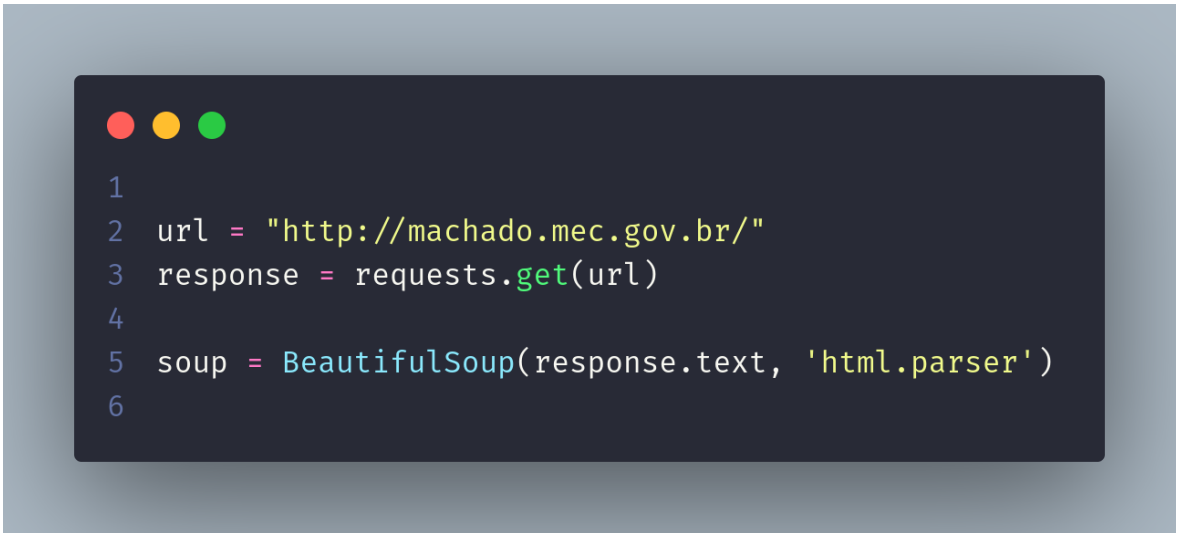
Ao clicarmos nesse item, somos levados a outra tela, que contém uma lista com todas as seguintes: romance, conto, poesia, crônica, teatro, crítica, tradução e miscelânea. Machado de Assis escreveu várias obras em cada uma dessas categorias. Assim, cada item desse nos leva a uma terceira tela, onde encontramos uma lista de todas as obras do autor de acordo com a categoria escolhida. A partir dessa listagem, iniciamos o processo de extração dos textos. A **Figura *24*** nos dá uma visão da tela de listagem de obras organizadas por categorias.

Figura 24 Listagem das obras organizadas por categoria

Fonte: http://machado.mec.gov.br/index.php

Iniciamos a construção do *script* definindo uma variável de nome URL. Nessa variável, armazenamos o endereço da página principal do *site*, ou seja, a URL[40], que foi o ponto de partida para a extração dos dados. A **Figura *25*** ilustra a definição dessa variável além de duas outras, *response* e *soup.* A primeira armazenou o resultado da chamada função *requests.get(url),* que retornou os dados da página em formato de texto, a segunda recebeu a transformação dos dados da página em componentes HTML.

Figura 25 Acessando a página principal



```
1
2  url = "http://machado.mec.gov.br/"
3  response = requests.get(url)
4
5  soup = BeautifulSoup(response.text, 'html.parser')
6
```

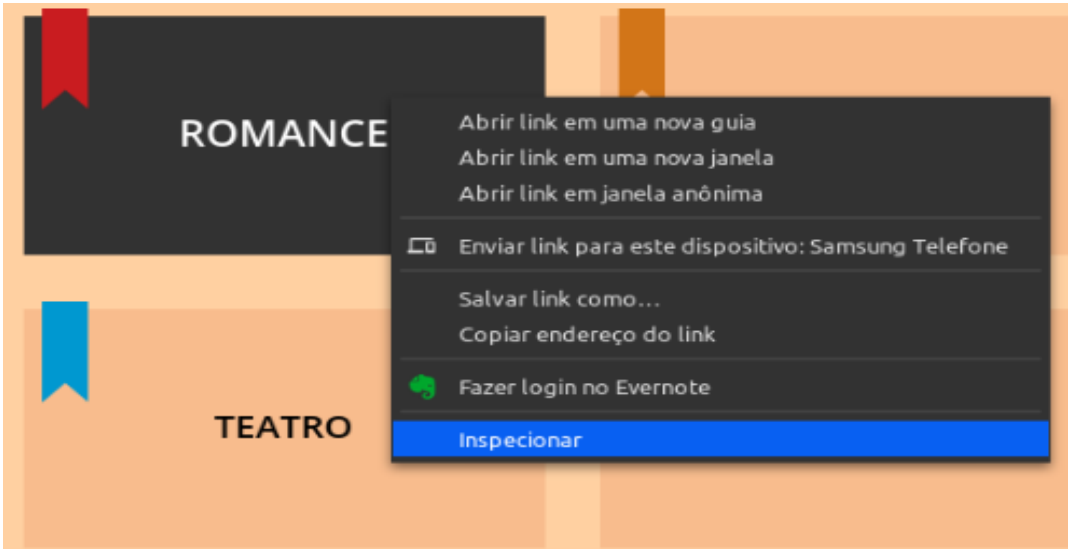
Fonte: O autor

40 URL é, basicamente, o endereço virtual de uma página ou de um website.

Conforme podemos observar na Figura 25, definimos como ponto de partida o endereço principal do *website*. A partir desse ponto, começamos a procurar os elementos na página que nos levaria à página que continha listagem com as obras organizadas por categoria.

Foi preciso inspecionar um dos elementos da listagem para entendermos como os elementos HTML's foram estruturados. Ao clicamos com o botão direito do *mouse* sobre um desses elementos, foi aberto um menu suspenso, e por meio dele, selecionamos a opção de inspecionar elemento, conforme é mostrado na **Figura** *26*. Após clicarmos em Inspecionar foi aberta uma aba no navegador, mostrando o código fonte da página do item que foi selecionado. Na **Figura 27**, ilustramos o código que foi utilizado para gerar a listagem das categorias.

Figura 26 Inspecionando o elemento



Fonte: http://machado.mec.gov.br/index.php

Figura 27 Inspecionando o item de menu - Romance

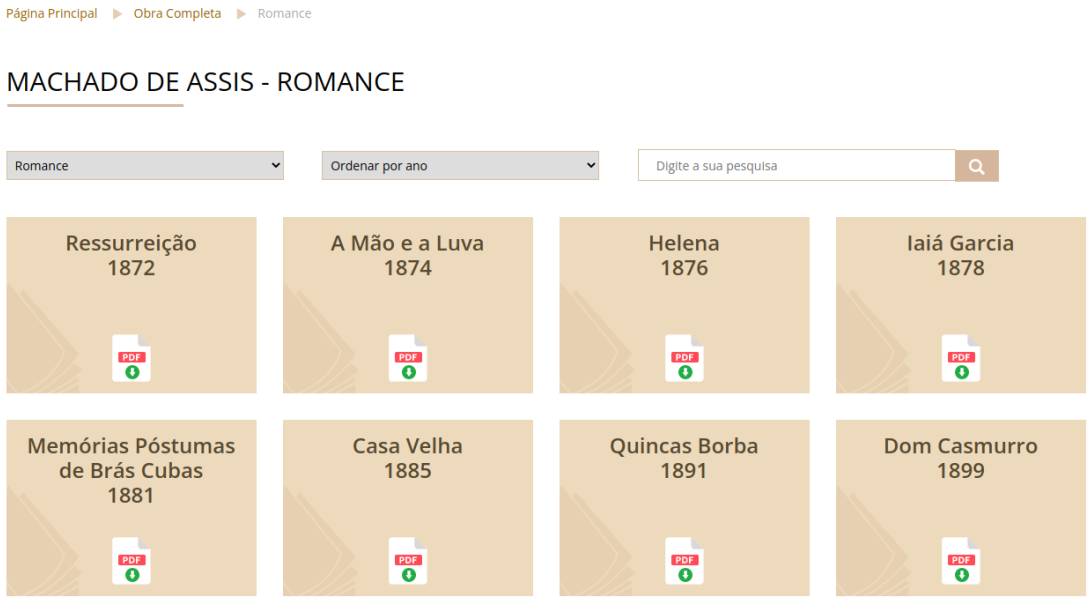
Fonte: O autor

Conhecendo como a página foi estruturada e como os elementos a configuram foram dispostos, começamos a ler esses dados. Para tanto, optamos por utilizar a biblioteca[41] de programação chamada *BeautifulSoup*. A escolha dessa biblioteca de programação se justifica por ser uma das mais utilizadas no mercado e por dispor de várias funcionalidades para manipulação de HTML. Quando o assunto é extração e manipulação de código HTML em *Python,* a biblioteca *BeautifulSoup* é uma das mais utilizadas e recomendadas pela comunidade acadêmica e pelos profissionais da área.

Todos os elementos que compõem os itens da listagem de categoria estão envoltos por um outro elemento "pai", que tem uma classe de definição de estilo chamada "*row* obras". Dentro de cada um desses elementos, podemos notar que temos um *link*, isto é, um elemento HTML, que tem a funcionalidade de redirecionar o usuário para outra tela ou, até mesmo, para outro *site*, a depender da URL para o qual ele está apontando. Nesse caso, ele nos levou para uma página com a listagem das obras daquela determinada categoria. Vejam, na **Figura *28***, a tela para a qual fomos encaminhados assim que clicamos na categoria.

41 São trechos de código que são desenvolvidos para realizar uma determinada tarefa em específico.

Figura 28 Tela com a lista de obras da categoria Romance



Fonte: http://machado.mec.gov.br/index.php

Demonstramos, na **Figura** 29, a seguir, o código escrito na linguagem de programação *Python* que percorreu todos os elementos HTML da tela e extraiu, de cada um deles, o endereço do arquivo em PDF da obra.

Figura 29 Percorrendo os elementos categorias

```
for obras in soup.find_all(class_='row obras'):
        pages_of_obras = [a['href']
                          for a in obras.find_all('a', href=True) if a.text]

        for page in pages_of_obras:
            # Só interessa para nos os nomes que não terminam com obra-completa-lista
            if not page.endswith('/obra-completa-lista'):
                url_to_obras.append("http://machado.mec.gov.br" + page)

```

Fonte: O autor

Conforme pode ser observado, criamos uma estrutura de repetição que percorreu e encontrou todos os elementos que estavam dentro do elemento “pai”, que contém a classe de nome "row obras". Para cada elemento que foi encontrado, buscamos por uma propriedade chamada "*href*". O valor atribuído a essa propriedade foi o endereço para a página que abarcava a lista das obras por categoria. Por exemplo: o componente intitulado

Romance tem um endereço para as obras dessa natureza. Observem um exemplo na **Figura *30***:

Figura 30 Elemento HTML categoria

```
▼<div class="row obras"> == $0
  ::before
 ▼<div class="col-md-3">
  ▼<div class="obra romance">
   ▶<div class="ribbon">…</div>
   ▶<a href="/obra-completa-lista/itemlist/category/23-romance">…</a>
    <div class="fundo-livro"></div>
   </div>
  </div>
```

Fonte: O autor

Na página com a listagem das obras por categoria, há a lista com todas as obras relacionadas àquela determinada categoria. Foi necessário, mais uma vez, inspecionar um dos elementos para entendermos como a lista com as obras foi estruturada, para conseguirmos localizar a URL do arquivo.

Figura 31 HTML com a listagem das obras

```
▼<div class="item"> == $0
 ▼<div class="detalhes">
    <div class="titulo"> Ressurreição </div>
    <div class="detalhe ano"> 1872 </div>
  </div>
 ▼<div class="download">
   ▶<a href="/obra-completa-lista/item/download/20_f90feea4579f3d4964f49e34dc473155" title="Download">…</a>
  </div>
 </div>
</div>
```

Fonte: O autor

Há dois itens importantes que fazem parte dessa estrutura: um é o componente que está agrupando os detalhes da obra como título e ano; o outro é o endereço para *download* do arquivo. Foi preciso extrair essas informações e compor o nome do arquivo e o endereço para *download*. Se observarmos, percebemos que o endereço para o arquivo não está completo, pois está faltando a parte inicial do endereço, que, nesse caso, é o domínio do *site.* A seguir, ilustramos o código que foi desenvolvido para realizar essa tarefa, que foi montar o nome e o endereço completo para *download* do arquivo.

Figura 32 Percorrendo as obras por categoria novo

```
for div in soup.find_all(class_="item"):
    for detalhes in div.find_all(class_="detalhes"):
        name = detalhes.find_all(class_="titulo")[
            0].text.strip().replace(" ", "_")

        if name.rfind('/'):
            name = name.replace("/", "_")

        if name.rfind('-'):
            name = name.replace("-", "a")

        ano = detalhes.find_all(class_="detalhe ano")[
            0].text.strip()

        if ano.rfind(' - '):
            ano = ano.replace(' - ', '_')

        filename = name + "_" + ano + ".pdf"

    get_file("http://machado.mec.gov.br/"+div.find("a",
                {"title": "Download"}).attrs['href'],
```

Fonte: O autor

No código da **Figura *32***, procuramos por todos os elementos HTML que tinha, a classe de nome “item”. Após encontrarmos esses elementos, percorremos essa lista e, concomitantemente, realizamos duas tarefas importantes: primeiro, acessamos o elemento com a classe de estilo "detalhes" e, de dentro dela, pesquisamos informações como o título e o nome da obra. Esses procedimentos foram necessários para que pudéssemos montar o nome da obra. Para isso, utilizamos o seguinte padrão: nome e ano de publicação dela. Ao compormos o nome, realizamos a substituição dos espaços em branco que separam o nome e o ano por *underline*[42]. Por exemplo, a composição final do nome foi: Ressurreição_1872.pdf.

Após realizar esses processamentos, chamamos uma função, que é um bloco de código que tem por responsabilidade realizar uma única tarefa, criada especificamente para realizar a tarefa de *download* do arquivo e, em seguida, passamos para essa função o caminho onde o arquivo se encontrava. Esse caminho foi formado pelo endereço juntamente com o nome da obra.

42 Esse termo é da língua inglesa e significa sublinhado.

Figura 33 Função que faz o *download* do pdf

```python
# Download file from url
def get_file(url, filename, category_of_obra):

    with requests.get(url, stream=True) as r:
        print("Download file: " + filename)

        path_corpus = 'src/corpus/pdf/'+category_of_obra+'/'
        if not os.path.exists(path_corpus):
            os.makedirs(path_corpus)

        with open(path_corpus+filename, 'wb', encoding='utf-8') as f:
            shutil.copyfileobj(r.raw, f)
            pdf2txt(filename, path_corpus, category_of_obra)

```

Fonte: O autor

Conforme pode ser observado na **Figura *33***, a função *get_file*[43] recebeu três informações importantes para o seu funcionamento, que foram: i) endereço do arquivo; ii) nome do arquivo; e iii) categoria à qual pertence o arquivo. A função iniciou o processo de *download* do arquivo, salvando-o em uma pasta específica.

A função responsável por fazer o *download* do arquivo em PDF ao final do seu processamento invocou outra função, chamada *pdf2txt*, que recebeu três informações importantes: i) nome do arquivo; ii) localização do *corpus*; iii) categoria a que pertence o arquivo. Essa função buscou pelo arquivo no *corpus*, com base no nome e no caminho do mesmo, e o converteu em arquivo texto, salvando o novo arquivo em uma pasta. Na **Figura 34**, temos uma ilustração da função que converteu arquivos em PDF para TXT.

43 Nome dado a função que faz o *download* do arquivo em PDF.

Figura 34 Função que converte o PDF em arquivo texto

```python
def pdf2txt(filename, path_corpus, category_of_obra):
    msg = "Converting {file} to txt: ".format(file=filename)
    print(msg)

    pdf_file_path = path_corpus+filename
    with open(pdf_file_path, 'rb') as file:
        pdf = pdftotext.PDF(file)

    if filename.endswith('.pdf'):
        path_corpus = 'src/corpus/txt/'+category_of_obra+'/'

        if not os.path.exists(path_corpus):
            os.makedirs(path_corpus)

        new_filename = path_corpus+filename.replace('.pdf', '.txt')

    with open(new_filename, 'w', encoding='utf-8') as output_file:
        for page in pdf:
            output_file.write(page)
```

Fonte: O autor

Foram apresentados, no capítulo, os procedimentos que utilizamos para automatizar a tarefa de obtenção e de conversão dos dados. Os procedimentos descritos demonstram o quanto as ferramentas computacionais podem facilitar o nosso trabalho e nos poupar de esforços, às vezes, desnecessários. Os ganhos com esse tipo de abordagem são em ordens de grandeza maior que procedimentos manuais. Ao fazermos o uso da programação para automatizar as tarefas mais repetitivas, o pesquisador economiza tempo e esforço, com isso ganha tempo para se dedicar ao que realmente importa, ou seja, realizar a análise dos dados.

Em uma pesquisa científica, por que não automatizar tarefas como coleta, nomeação de arquivo, salvamento de arquivos, disponibilização, conversão, limpeza e normalização dos dados? Fatores como a repetição excessiva de determinada tarefa, a quantidade de procedimentos que temos que fazer para executar essa tarefa e o número de vezes que a executamos são indícios que devemos levar em consideração para a busca de alternativas que possam automatizá-las, deixando a cargo do computador executar esse tipo de trabalho.

Segundo Berber Sardinha (2004, p. 72), uma vez que os textos tenham sidos coletados e limpos, a tarefa seguinte é a organização dos arquivos em uma estrutura coerente. Tentando seguir essa orientação, dividimos a organização dos arquivos em uma estrutura de pastas em que cada pasta indicou quais textos estava armazenados dentro dela.

Apresentamos a seguir, de forma pontual e detalhada, quais foram os procedimentos adotados para que fosse possível utilizar um *script* que automatizasse o processo de *download* dos textos que compuseram o *corpus*.

3.2.8 Importação do *corpus*

Após a realização das adequações estruturais necessárias no sistema, começamos a construir o banco de dados popular com os textos, mas antes de iniciarmos a importação, foi preciso efetuar alguns passos importantes. As obras que utilizamos estavam disponibilizadas no domínio público em formato PDF[44], mas esse não era o formato de arquivo mais indicado para utilizarmos na análise exploratória de textos, sendo assim, precisamos converter esses textos em formato PDF para o formato TXT[45]. Arquivos nesse formato não possuem nenhum tipo de formatação e/ou estilização, pois são textos ditos crus (*plain texts ou raw data)*[46].

Para a realização dessa tarefa, era possível utilizar diversas ferramentas disponíveis na *internet* com opções de ferramentas gratuitas e pagas. O *site pdfcandy* converte vários formatos de arquivos para vários outros formatos. No nosso caso, conforme foi demonstrado anteriormente, nós optamos por fazer essa tarefa logo após o *download* do arquivo, por meio de uma função escrita em *Python*. Nosso *script* fez o *download* do arquivo em PDF e a conversão automática para TXT. De posse do texto já convertido, fizemos apenas algumas adequações.

Berber Sardinha (2004, p. 53) sugere a utilização de *scripts*[47] como uma forma de realização da limpeza dos textos. Para este trabalho, no entanto, não foi necessária a utilização deste tipo de abordagem, mas, para facilitar, nós utilizamos um comando que fez algumas substituições de palavras.

---

46 De acordo com Baker, Hardie e Mcenery (2006), *plain text* é um texto que contém somente palavras um documento original.

47 Trechos de código que são escritos para fazerem determinadas tarefas, normalmente são criados para executar uma única tarefa.

Figura 34 Comando para substituir as palavras

```
heitor in ~
❯ cat  Documentos/Desenvolvimento/web\ crallig/src/corpus/txt/romance/Dom_Casmurro_1899.txt | sed s/CAPÍTULO/CHAPTER/ > text.txt
```

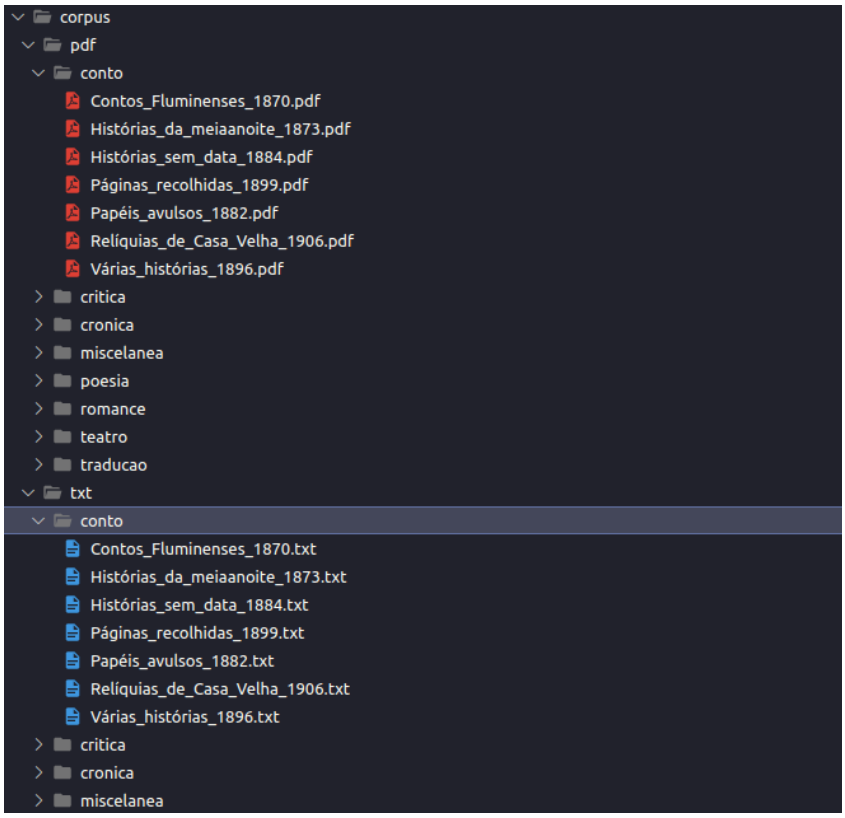
Fonte: o autor

3.2.9 Organização do *Corpus*

Logo após a etapa de *download* e conversão dos textos damos início a outra etapa que é muito importante, a organização dos textos, pois, não podemos deixar de dar a atenção necessária com relação à organização do *corpus.*

Assim, nesta pesquisa, organizamos o *corpus* da seguinte forma: uma pasta chamada *corpus*, local onde foram armazenados todos os arquivos; mais duas outras pastas, uma chamada PDF e outra TXT, ambas para armazenar os arquivos em formatos PDF e TXT, respectivamente. Vale ressaltar que o *corpus* ficou armazenado no disco rígido do computador pessoal do pesquisador, essa estrutura não foi armazenada no servidor onde o sistema foi hospedado. A **Figura *35*** ilustra como ficou a organização do *corpus* após o *download* e a conversão dos arquivos.

Figura 35 Organização do *corpus*



Fonte: o autor

Após as etapas de *download* e conversão dos textos o nosso *corpus* ficou organizado em uma estrutura de pastas, primeiramente os textos foram separados por formatos, TXT e PDF, e depois foram agrupados em pastas de acordo com a sua categoria. Ao final do processo o resultado foi uma estrutura em formato linear, partindo do formato do arquivo até o arquivo em si.

Na próxima sessão damos mais detalhes sobre o processo de preparação do *corpus,* quais procedimentos adotamos para preparar os textos para serem utilizados na plataforma.

3.2.10 Processo de preparação dos textos para importação

Após a obtenção dos textos, o pesquisador precisa se certificar de que eles são "úteis" para serem utilizados como um *corpus*. Cabe ao pesquisador se preocupar com a consistência dos dados e com a validação do trabalho que foi realizado pelo computador. A utilidade de um texto, para as pesquisas da LC, está associada, obrigatoriamente, à condição favorável dele para o processamento por meio de ferramentas computacionais.

Antes de dar início à importação propriamente dita dos textos, foi importante fazer uma conversão desses arquivos para que, posteriormente, não surgissem problemas de compatibilidade.

Quando criamos um arquivo em um determinado Sistema Operacional (SO), algumas características desse sistema são colocadas no arquivo, como por exemplo: quando criamos um arquivo de texto no Windows é adicionado ao final da linha um carácter especial para indicar quebra de linha. No Windows, o padrão de quebra de linha corresponde à combinação dos caracteres de controle Carriage Return (CR) "\r" e Line Feed (LF) "\n", ou seja, ao final de cada linha são adicionadas as seguintes sequencias de caracteres "\r\n", esses caracteres ficam ocultos no texto e são visíveis somente por um editor de texto.

Ao convertermos os arquivos de PDF para TXT, a quebra de linha veio em um formato que é amplamente utilizado pelo sistema operacional Microsoft Windows. Os arquivos em formato que o Windows reconhece exigem que as linhas terminem com os seguintes caracteres: "\n\r", conforme dito anteriormente. Esses caracteres indicam uma quebra de linha e são invisíveis para nós seres humanos, mas são imprescindíveis para

um correto funcionando do sistema operacional. Em sistemas baseados em Unix[48], a representação de uma nova linha é feita apenas com o carácter Line *Feed* (LF) "\n". Como o sistema foi desenvolvido para rodar em sistema operacional Linux, que é uma variante do *Unix*, foi preciso então fazer a conversão dos arquivos. Sendo assim, para realizarmos essa conversão, utilizamos o *software* chamado *dos2unix*, que é um utilitário de linha de comando.

O *dos2unix*[49] é um *software* que foi desenvolvido para ser executado em um terminal de linha de comando. Para iniciá-lo, devemos acessar o *software* interpretador de comandos chamado de terminal. Na **Figura *36***, temos um exemplo de como fizemos para utilizar o *dos2unix*. Nessa linha de comando, convertemos todos os arquivos da pasta **Romance** para o formato TXT, qual precisávamos para dar andamento nas próximas etapas.

Figura 36 Comando utilizado para converter arquivos TXT

```
heitor in ~
› for f in corpora/Machado/ROMANCE/*.txt; do dos2unix -m $f; done
```

Fonte: O autor

Com a utilização desse comando, buscamos por todos os arquivos que estava dentro da pasta **Romance** que tem a extensão TXT e, em seguida, convertemos cada um deles para o formato correto, como resultado desse processo obtivemos todos os arquivos dessa pasta em um padrão correto para ser utilizado nas próximas etapas.

Outra etapa importante é a de limpeza dos textos, nela removemos ruídos nos textos. A seguir, mostramos como foi realizada a limpeza dos textos.

3.2.11 Limpeza do *corpus*

A fim de que os textos possam atingir a condição desejada para serem importados (Formato TXT, limpos e padronizados), é necessária a realização de uma série de procedimentos, dentre eles, o processo de limpeza, que é um dos mais importantes.

A limpeza e a normalização dos textos de um *corpus* são maneiras de reduzir ou eliminar erros que possam ter ocorrido durante o processo de conversão ou, até mesmo,

48 Unix é um sistema operacional criado por Kennedth Thompson na década de 60 nos laboratórios da Bell e AT&T nos Estados Unidos da América.
49 Ferramenta de linha de comando utilizada para converter arquivos para sistema operacionais baseados em Unix, para que possa entendê-los.

durante o processo de obtenção dos textos. De acordo com Aluísio e Almeida (2006), a limpeza do *corpus* consiste na remoção dos ruídos linguísticos, na correção de erros e na padronização de dados linguísticos.

A revisão de erros, geralmente, está ligada às palavras que passaram a apresentar a grafia incorreta após a conversão do texto para o formato TXT, e a normalização de dados linguísticos abrange a uniformização de palavras, de siglas e de abreviaturas, que possuem variações de escrita, a remoção de espaçamentos e de quebras de linhas desnecessárias e a homogeneização de caracteres de pontuação do texto, como hifens, traços, aspas e apóstrofos.

No Quadro 5, a seguir, listamos alguns dos procedimentos que são necessários para essa finalidade.

Quadro 5 - Procedimentos para limpeza e normalização

Remoção de cabeçalhos e rodapés de páginas.
Remoção de elementos gráficos (figuras, imagens e gráficos).
Remoção de notas de rodapé e fim.
Remoção de números de página.
Remoção de referências bibliográficas.
Remoção de listas (sumários, figuras, abreviações e gráficos).
Remoção de tabelas e quadros.
Remoção de títulos e subtítulos.
Remoção de qualquer texto de prefácio atribuído a uma pessoa que não seja o autor.

Fonte: o autor

Infelizmente, para essa etapa do processo não foi possível escrever um *script* que fizesse essa automação, pis são itens que necessitam de verificação humana como: correção de acentuação, correção de palavras, remoção de espaços em branco etc.

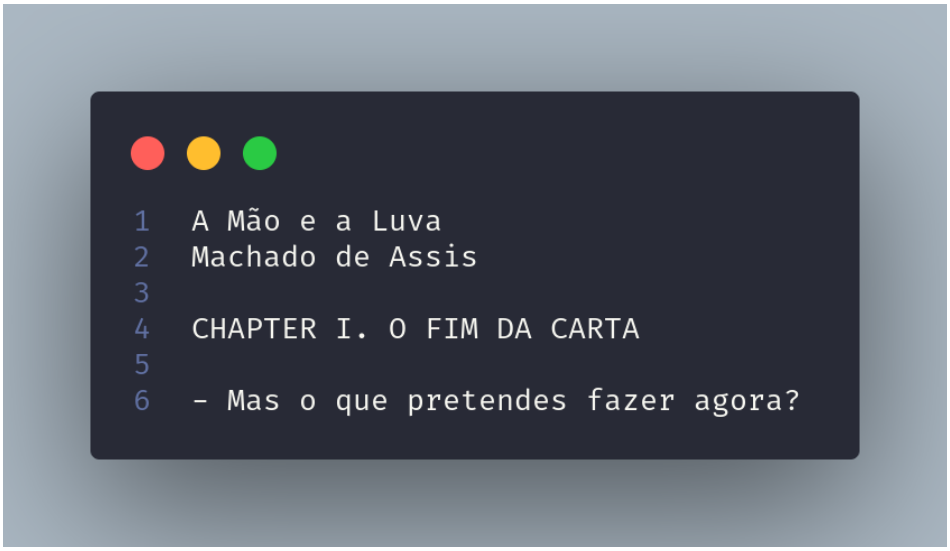
Abaixo apresentaremos o formato dos textos e, tratamentos que demos a alguns itens dos textos, como os títulos.

3.2.12 Formatação do texto

No Quadro 1, listamos os itens que deveriam ser removidos dos textos, entretanto, ainda precisávamos que fossem feitas mais algumas adequações. Após esses procedimentos de limpeza, uma das primeiras e mais importantes adequações que fizemos foi colocar na primeira linha de cada texto o nome da obra e, logo em seguida, na segunda

linha, o nome do autor, assim como é mostrado na **Figura** 37. Foi necessário fazer esse procedimento para que o sistema identificasse o autor e a obra.

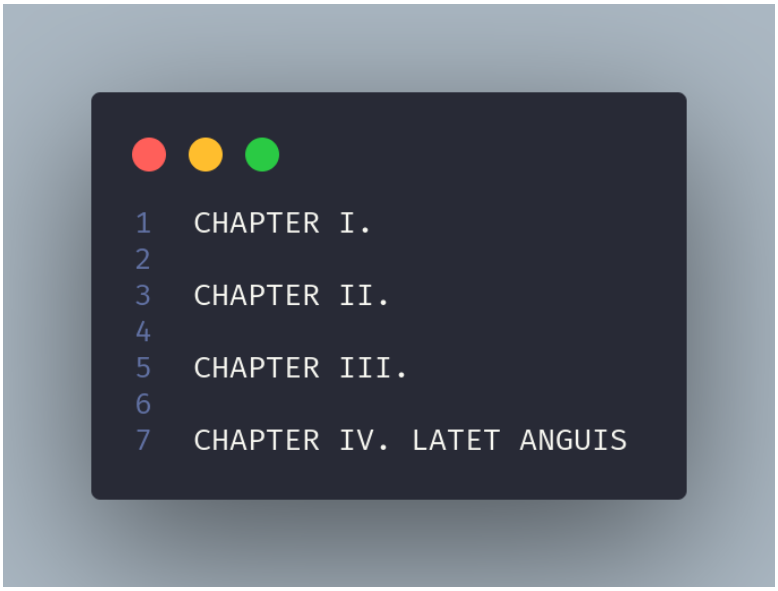
Figura 37 Definindo o nome da obra e do autor



Fonte: O autor

Com essa etapa da adequação concluída, renomeamos os capítulos dos livros. O título do capítulo começou com “CHAPTER” ou “BOOK”, e foi seguido por número ou por algarismos romanos e um ponto. O número do capítulo ou do livro não pôde ser escrito em forma de palavra. O título foi seguido opcionalmente pelo título do capítulo, o qual não quebrou em uma nova linha. Na **Figura *38***, estão alguns exemplos de como os capítulos ficaram formatados.

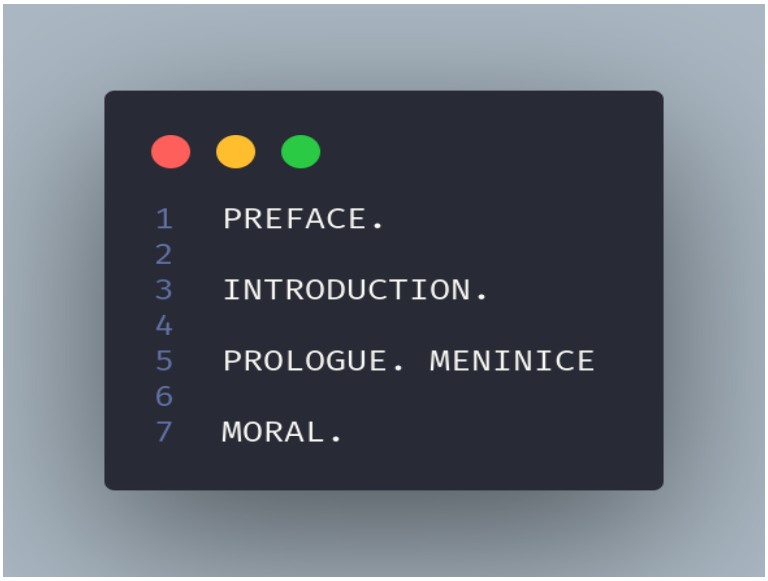
Figura 38 Renomeando os capítulos



Fonte: O autor

Ainda, seguindo as adequações do texto, temos os títulos, prefácios, conclusão, prólogo, prelúdio ou moral. Esses itens foram tratados como os capítulos. Convém esclarecer que eles não exigiram números, mas o ponto. Novamente, o título foi seguido opcionalmente por um título, que não quebrou em uma nova linha. Na **Figura 39** estão elencados alguns exemplos.

Figura 39 Renomeando os títulos, os prefácios e a introdução



Fonte: O autor.,

Ainda nessa fase de preparação e formatação do texto, tivemos um passo igualmente importante aos que já havíamos executado até o momento, que foi identificar no texto onde começava e onde terminava cada sentença e anotar essa informação em um arquivo. A esse processo demos o nome de sentenciamento. Essa atividade gerou, a partir de um arquivo de texto de entrada, um outro arquivo em formato CSV[50], que continha os pontos de início e de fim de cada sentença, juntamente com a própria sentença e um metadado, que classificou essa informação em título, nome do autor, parágrafo ou sentença.

Como uma forma de automatizar esse processo, foi disponibilizada uma ferramenta de linha de comando chamada *region_export*. Essa ferramenta recebeu como parâmetro o endereço do arquivo texto que desejávamos realizar o processamento. Na

50 csv (comma separated values) é um tipo de arquivo de texto cujo conteúdo é separado por vírgulas.

Figura 40, demonstramos como executamos essa ferramenta. Para executá-la, fizemos o seguinte caminho: “corpora/Machado/*/*.txt”. O primeiro asterisco indica para a ferramenta que é ela que deve percorrer dentro de qualquer pasta que esteja dentro da pasta Machado; o segundo indica que é ela que deve ler todos os arquivos que tenha a extensão .txt, independentemente do nome.

Figura 40 Executando a ferramenta *region_export*

Fonte: O autor

Ao final desse processo, a ferramenta gerou no mesmo local do arquivo .TXT que ela leu um novo arquivo com o nome do arquivo de origem juntamente com a terminação *.regions.csv.* Nesse arquivo, obtivemos todos os metadados do arquivo de entrada, conforme pode ser visto na **Figura 41**.

Figura 412 Meta dados gerados pela ferramenta region_export

	Padrão	Padrão	Padrão	Padrão	Padrão
1	metadata.title	0	7		Helena
2	metadata.author	8	24		Machado de Assis
3	chapter.title	30	40	1	CHAPTER I.
4	chapter.text	42	10332	1	O Conselheiro Vale m... rece
5	quote.nonquote	42	10332		O Conselheiro Vale m... rece
6	chapter.paragraph	42	484	1	O Conselheiro Vale m...te fo
7	chapter.sentence	42	111	1	O Conselheiro Vale m...25 de

Fonte: O autor

Antes de darmos andamento com a importação das obras, tivemos que nos certificar de que o processo de sentenciamento dos textos estava correto, isto é, foi preciso verificar se a ferramenta *region_export* conseguiu de fato identificar todos os atributos do texto e gerar corretamente o arquivo.*regions.csv.* Caso a ferramenta não conseguisse identificar uma sentença quando fôssemos proceder com a análise do texto no sistema, poderíamos encontrar algumas inconsistências.

Para assegurarmos que tudo estava correto, tivemos à nossa disposição uma outra ferramenta também de linha de comando, o *region_preview*, diferentemente do

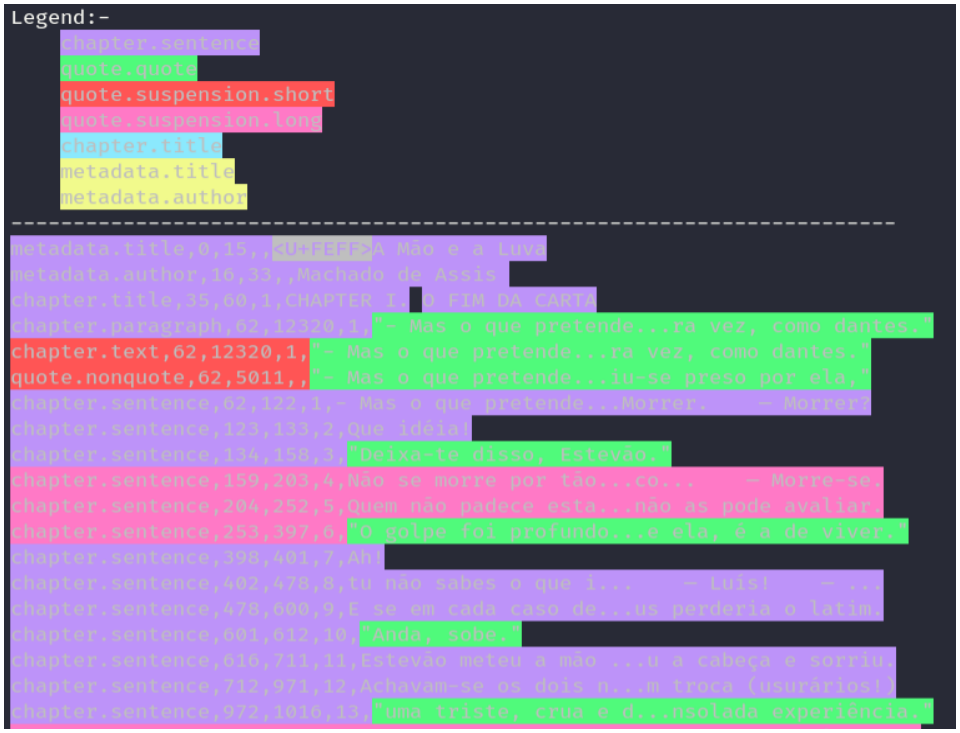
region_export, o qual passamos como argumento um arquivo em formato .TXT, para o *region_preview* passamos o arquivo com a extensão *regions.csv*, conforme ilustrado a seguir.

Figura 43 Executando a ferramenta *region_preview*

Fonte: O autor

Por meio dessa ferramenta, podemos visualizar graficamente, com base em legendas e em cores, o que foi identificado no texto.

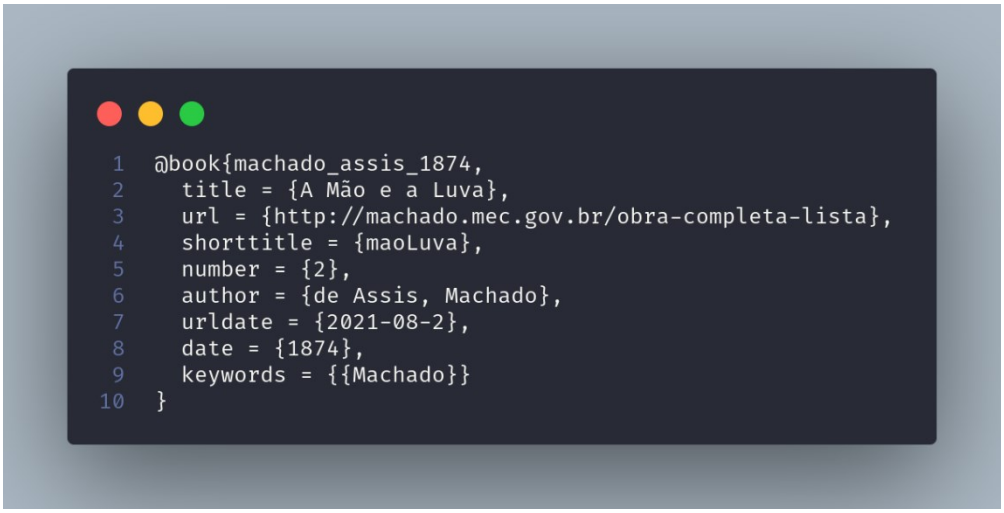
Figura 44 Visualizando o resultado da ferramenta *region_preview*



Fonte: O autor

Houve ainda mais um passo que realizamos para, enfim, fazermos a importação do *corpus*. Foi preciso criar na pasta raiz do *corpus* um arquivo com as anotações dos textos que estávamos importando. Esse arquivo foi uma espécie de dicionário, nele anotamos os metadados dos textos que estávamos importando, como: *title, url, author*. Na **Figura 42**, apresentamos um exemplo de como elaboramos esse arquivo.

Figura 42 Conteúdo do arquivo *corpora.bib*

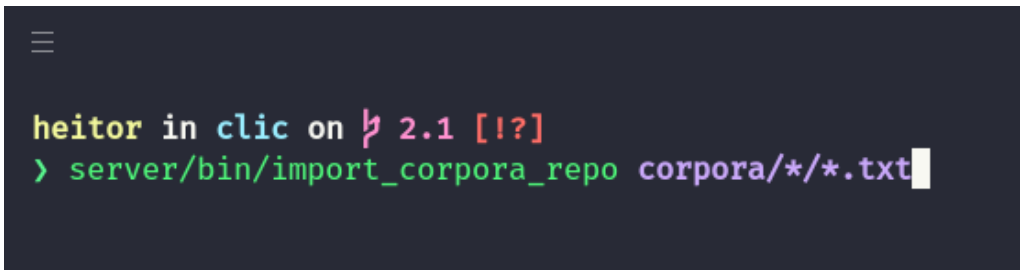


```
@book{machado_assis_1874,
  title = {A Mão e a Luva},
  url = {http://machado.mec.gov.br/obra-completa-lista},
  shorttitle = {maoLuva},
  number = {2},
  author = {de Assis, Machado},
  urldate = {2021-08-2},
  date = {1874},
  keywords = {{Machado}}
}
```

Fonte: O autor

Com todos os procedimentos adotados, estávamos aptos a realizar a importação das obras e, para isso, foi preciso utilizar outro utilitário de linha de comando. Neste caso, usamos não só o *import_corpora_repo,* o *region_export*, informando o caminho dos arquivos, mas também passamos para ele o caminho dos textos em formato .txt.

Figura 43 Executando a ferramenta import_corpora_repo



```
heitor in clic on  2.1 [!?]
❯ server/bin/import_corpora_repo corpora/*/*.txt
```

Fonte: O autor

Assim que executamos o comando, ilustrado na **Figura 43**, o processo de importação das obras se iniciou. Esse procedimento foi um pouco demorado, pois o tempo de processamento dependia da quantidade de obras que seriam inseridas, do tamanho de cada obra e do computador onde o sistema foi instalado.

A partir do próximo capítulo, iniciamos a fase de análise das funcionalidades do sistema, dando mais detalhes sobre cada um dos módulos do sistema *GEConWeb*. Apresentamos uma visão geral sobre o sistema e as suas funcionalidades, exemplificamos por meio de imagens como os recursos podem ser acessados e utilizados e com isso conseguimos extrair da ferramenta alguma informação relevante sobre os *corpora*.

ANÁLISE DOS MÓDULOS E FUNCIONALIDADES DO SISTEMA

Para o desenvolvimento do sistema, optamos por utilizar um *layout*[51] simples, de forma a facilitará a identificação e o uso dos recursos que a plataforma iria prover. O *layout* foi pensado e estruturado de uma forma que fosse possível atender aos projetos como um todo, e que fosse mantida uma identidade visual entre os trabalhos. A ideia foi causar no usuário uma sensação de uniformidade e coerência entre eles, permitindo assim uma melhor usabilidade da ferramenta.

A interface é um dos elementos mais importantes em um projeto de *software*, pois esse é o ponto de entrada com o qual o usuário tem o seu primeiro contato com a plataforma. É por meio da interface que o usuário tem acesso aos recursos que são disponibilizados por ela. Usabilidade é sinônimo de facilidade de uso, se o uso de um produto é fácil e intuitivo, o usuário tem maior produtividade: aprende mais rápido a utilizá-lo, memoriza as operações e comete menos erros. (MANDEL, 1997)

Abaixo temos uma figura que listra a página inicial do *GEConWeb*.

Figura 44 Página inicial do sistema GEConWeb

Fonte: https://www.ileel.ufu.br/gecon/

A página inicial da ferramenta também é a página inicial do grupo de pesquisa GECon – Grupo em Estudos Contrastivos, presidido pelo prof. Doutor Ariel

51 Layout se refere à disposição visual dos elementos em um documento, página *web*, aplicativo ou qualquer outro tipo de produto gráfico ou digital.

Novodvorski. Aproveitamos a página inicial para apresentarmos ao público o grupo de estudos, mostrar qual é o seu objeto de estudo e quais as suas áreas de atuação.

A seguir, detalhamos cada um dos módulos bem como as suas funcionalidades.

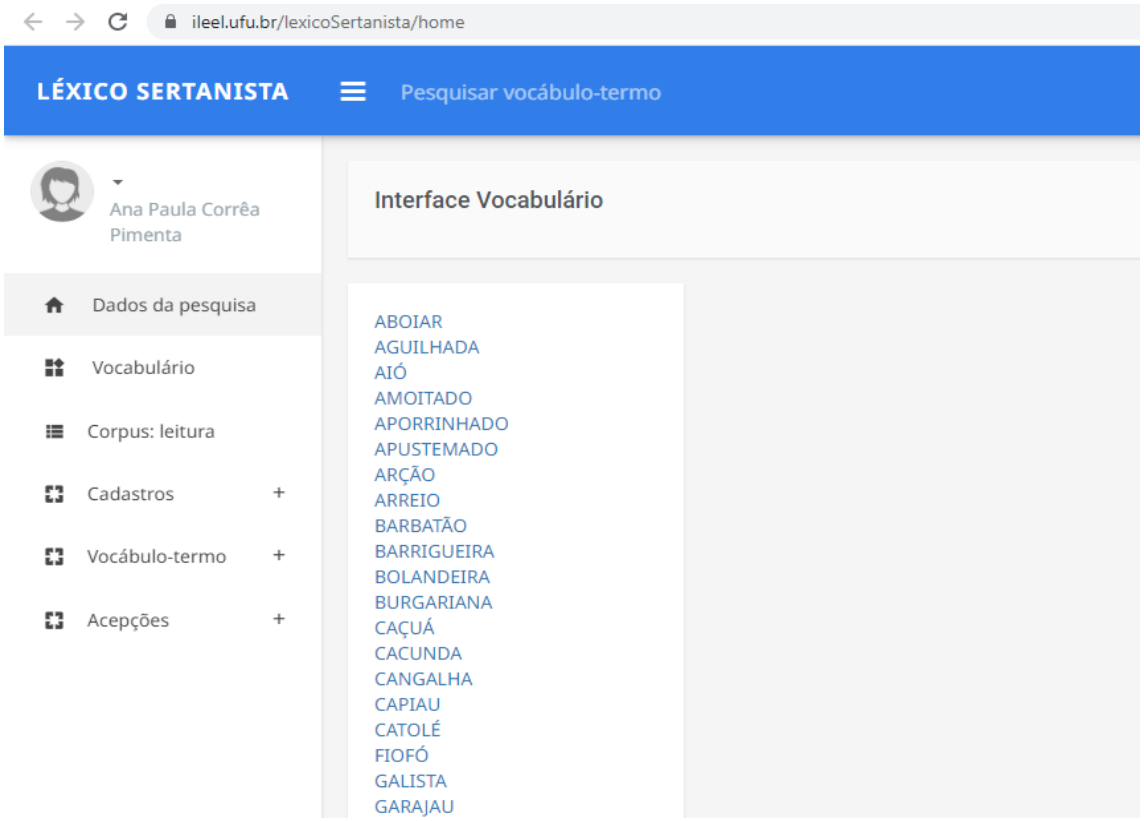
4.1 LÉXICO SERTANISTA

Apresentamos as seguir as funcionalidades desenvolvidas para o módulo Léxico Sertanista. Aplicamos o conceito de usabilidade de *software* no sistema de **Vocábulo**, termo que foi desenvolvido para a pesquisa de Pimenta (2019). Esse conceito nos guiou durante o desenvolvimento dos módulos da ferramenta. Esse modulo integrará a plataforma que disponibilizaremos para análise de *corpus* online como mais uma forma de visualizar e manipular um *corpus*.

Pensando na acessibilidade e na usabilidade do sistema, decidimos disponibilizar o acesso às informações por meio de dois módulos: uma é o modo **Leitura** e o outro é o modo **Vocabulário**. No modo **Leitura**, é possível localizar todas as obras integrantes do *corpus* de estudo; já no modo **Vocabulário**, teremos uma listagem com os vocábulos-termos, suas acepções e as obras das quais ele foi retirado. Esses dois modos, além de servirem como modelo para o desenvolvimento de futuras ferramentas de visualização, oferecem facilidades e possibilidades de consulta, pela rapidez do acesso, ou seja, pela forma como os dados são apresentados. Por essa razão, optamos por esse formato.

Abaixo apresentamos a página inicial o Léxico Sertanista com alguns vocábulos termos que foram obtidos pela pesquisadora (Pimenta, 2009).

Figura 45 Home Léxico Sertanista



Fonte: Pimenta (2009)

Por meio do modo **Leitura,** podemos acessar as obras que foram separadas por fases do regionalismo, sendo elas, respectivamente: Regionalismo Pitoresco (Fase 1); Regionalismo Crítico (Fase 2); e Super Regionalismo (Fase 3). Ao selecionarmos uma das fases que compõem o menu lateral esquerdo, temos, do lado direito, uma lista com as obras da fase selecionada, conforme pode ser visto na figura abaixo.

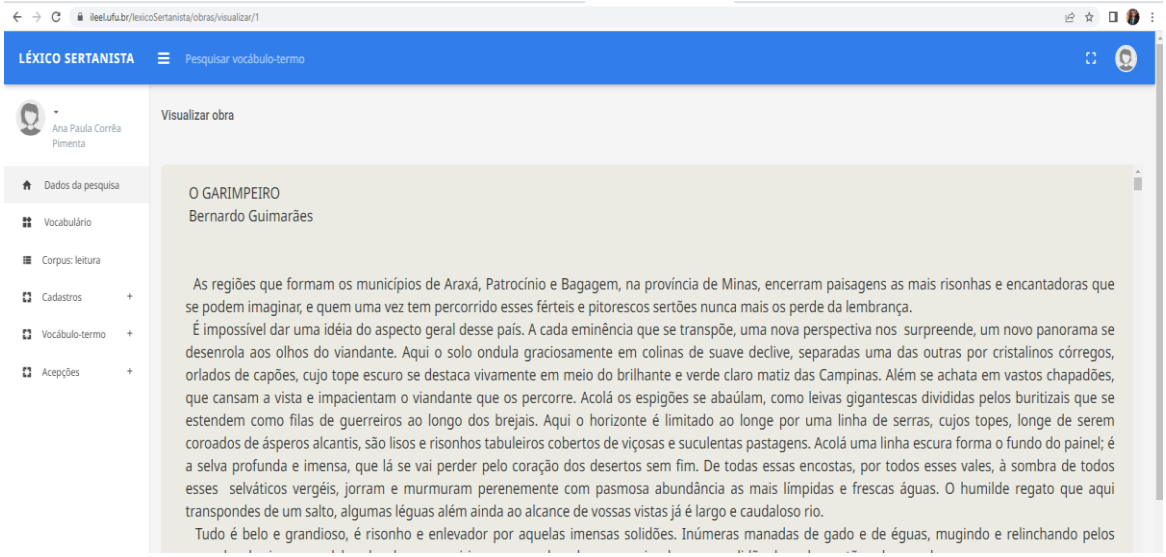
Figura 46 Modo leitura

Fonte: Pimenta (2009)

Na tela ilustrada na **Figura** 46, temos alguns dados das obras com base na fase do regionalismo que foi selecionada, tais como: nome da obra, nome do autor, sigla, data em que a obra foi publicada. Além dessas informações, temos duas outras opções: a primeira é a possibilidade de fazer o *download* da obra, a segunda é a possibilidade de ler a obra como um todo, sendo essa última uma forma de termos acesso ao texto da obra.

No modo **Leitura**, ilustrado pela **Figura** 47, temos acesso à obra por completo. Ao longo da leitura, podemos encontrar algumas palavras (verbetes) que estão em destaque no texto, em cor diferente. Quando encontramos um verbete e clicamos nele é aberto um *pop-up*: uma caixa com informações sobre esse verbete. Dentro desse *pop-up* temos informações do contexto abonatório de onde ele foi retirado, a classificação gramatical, a frequência com que esse verbete aparece no *corpus*, dentre outras informações. A **Figura** 48, a seguir, mostra o *pop-up* [52]com as informações do verbete.

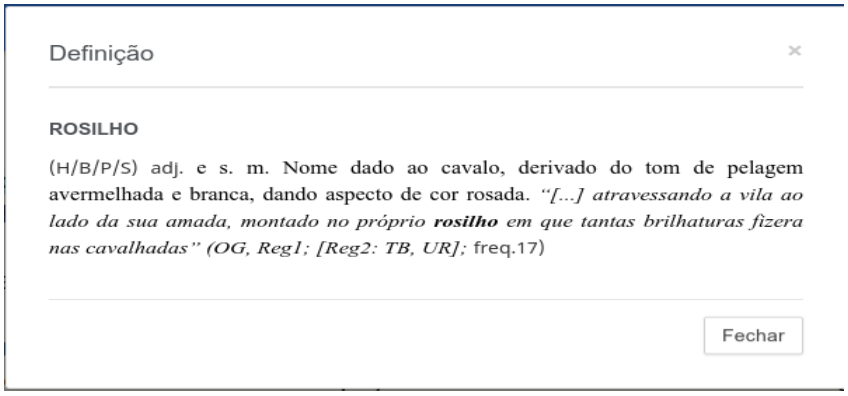
Figura 47 Modo leitura do Léxico Sertanista



Fonte: Pimenta (2019)

52 Significado de *Pop-up*. Disponível em: https://www.significados.com.br/pop-up/#:~:text=Pop%2Dup%20%C3%A9%20uma%20janela,dos%20casos%2C%20publicidades%20e%20an%C3%BAncios. Acesso em: 12 jul. 2021

Figura 48 Pop-up do verbete



Fonte: Pimenta (2019)

Já no modo **Vocabulário,** temos acesso há uma listagem com todos os **vocábulos-termo** do *corpus* classificados por ordem alfabética. Além disso, está disponível nesse modo de apresentação um campo que nos permite fazer pesquisas por um **vocábulo-termo** em específico. Quando iniciamos a digitação do termo que desejamos pesquisar no campo "Pesquisar vocábulo-termo", caso o termo conste no *corpus,* aparecerá uma caixa com a sugestão do termo que foi localizado, conforme pode ser visto na **Figura 49**.

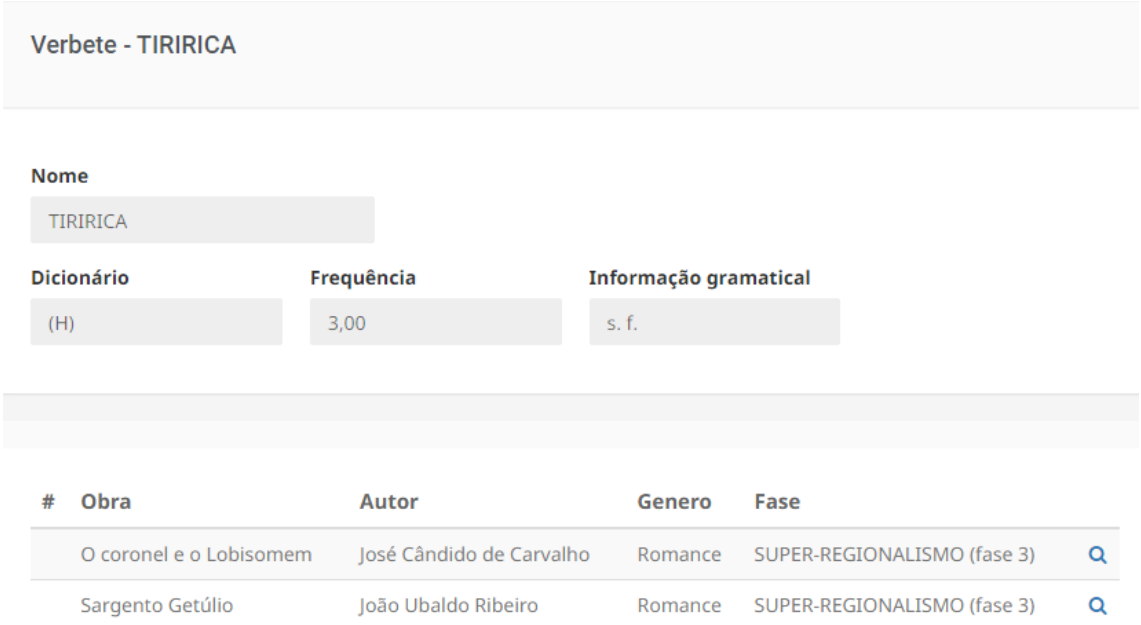
Figura 49 Modo pesquisa rápida

Fonte: Pimenta (2019)

Ao clicarmos no **vocábulo-termo**, que é apresentado no campo suspenso da pesquisa rápida, somos direcionados para uma tela com informações a respeito do termo

encontrado. Nessa tela, temos acesso a alguns dados como: nome, frequência, informação gramatical além de termos uma lista com a relação de obras onde esse verbete aparece. A seguir, na **Figura 50**, apresentamos uma ilustração dessa tela, sendo que em cada uma das obras listadas há um botão em formato de lupa que nos direciona para uma segunda tela, onde nos é mostrada a obra em que esse verbete foi encontrado. Nesta tela, há a possibilidade de ler a obra por completo; nela encontramos o **vocábulo-termo** pesquisado em destaque, marcado com uma cor de fonte diferente. Essa marcação é feita para que possamos identificá-lo mais facilmente.

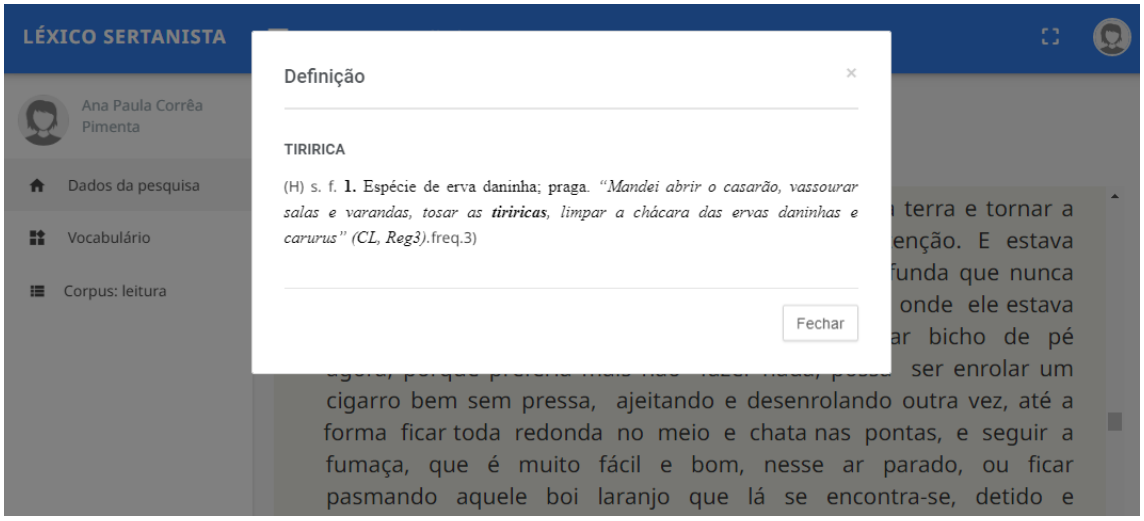
Figura 50 Detalhes do verbete



Fonte: Pimenta (2019)

Ainda nessa tela, que chamamos de modo leitura da obra, quando identificamos um verbete e temos a curiosidade de saber o seu significado, podemos clicar nele e, assim, será aberto um *pop-up* com algumas das suas informações, como significado e informações gramáticas, como pode ser observado na **Figura 51**.

Figura 51 Modo leitura com pop-up

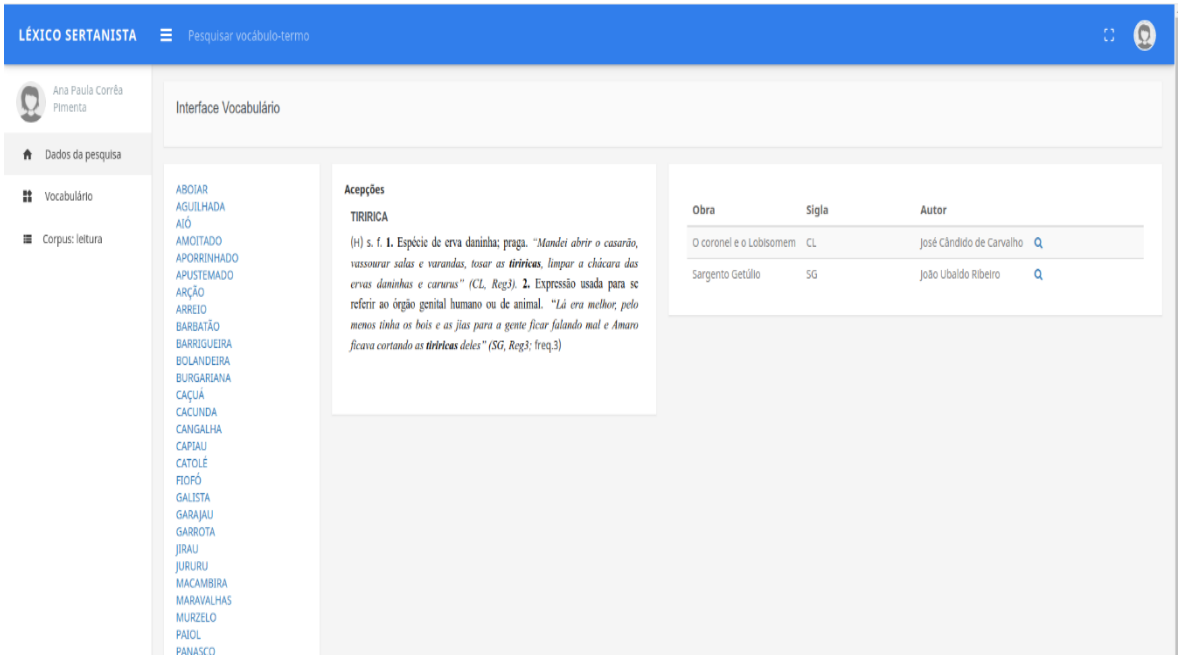


Fonte: Pimenta (2019)

Podemos observar ainda no modo **Vocabulário** que, ao clicarmos em um **vocábulos-termo**, por exemplo: "TIRIRICA", na mesma tela do lado direito aparecem os detalhes e os textos em que o **vocábulo-termo** foi encontrado. Desse modo, conseguimos disponibilizar para o consulente de uma forma rápida e prática o acesso ao maior número de informações de um **vocábulos-termo,** concentradas em um único lugar. Criamos também uma opção para que o consulente possa ter acesso às obras com os **vocábulos-termos** e suas definições por meio de *pop-ups,* bastando que ele clique na lupa ao lado do nome do autor da obra.

A **Figura 52** sintetiza toda essa tela em uma única figura, com os detalhes do **vocábulo-termo** TIRIRICA selecionado.

Figura 52 Modo vocabulário com detalhes do verbete



Fonte: Pimenta (2019)

A seguir, apresentamos os detalhes do módulo intitulado como Léxico Indianista.

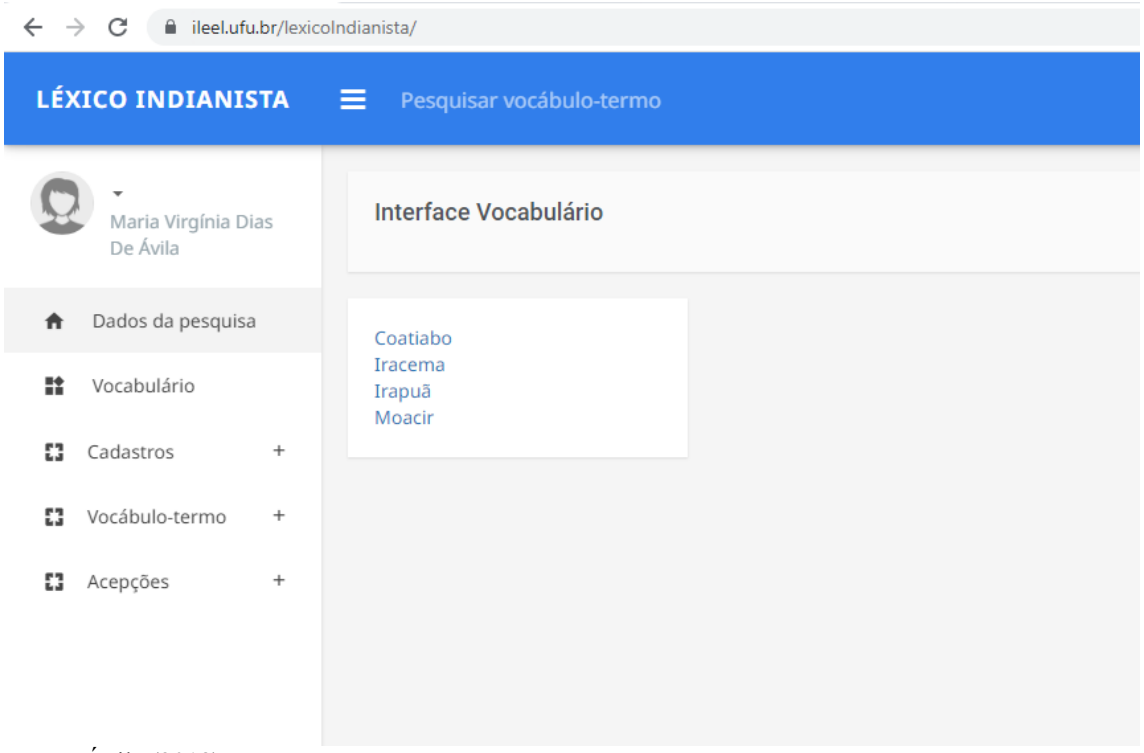
4.2 LÉXICO INDIANISTA

Por compartilhar uma mesma plataforma de desenvolvimento, os módulos da plataforma *GEConWeb* seguiram em uma mesma linha de *design*[53], portanto, as páginas podem apresentar algumas semelhanças entre sei, porém, com informações e funcionalidades distintas.

No caso, o Léxico Indianista segue na mesma linha do Léxico Sertanista, haja vista que as formas de visualização são bem parecidas, mas com alguns detalhes. Os dois trabalhos visam demonstrar o emprego do verbete nas obras dos autores, dando ênfase o seu uso. Mostramos abaixo a página inicial do Léxico Indianista e, demonstramos a disposição das informações na tela.

53 É a idealização, criação, desenvolvimento, configuração e elaboração de produto, tanto de *software* quando industrial. Disponível em: https://pt.wikipedia.org/wiki/Design. Acesso em: 14 nov. 2022.

Figura 53 Página inicial do modulo Léxico Indianista



Fonte: Ávila (2018)

Temos na tela inicial alguns dos verbetes que foram levantados pela pesquisadora, eles foram recolhidos das três obras que compõem o *corpus* do léxico indianista: "Iracema", "O Guarani" e "Ubirajara". Quando escolhemos um verbete e clicamos sobre ele nesta página, nos são apresentadas algumas informações relacionadas ao verbete escolhido, como: classe gramatical, significado e frequência.

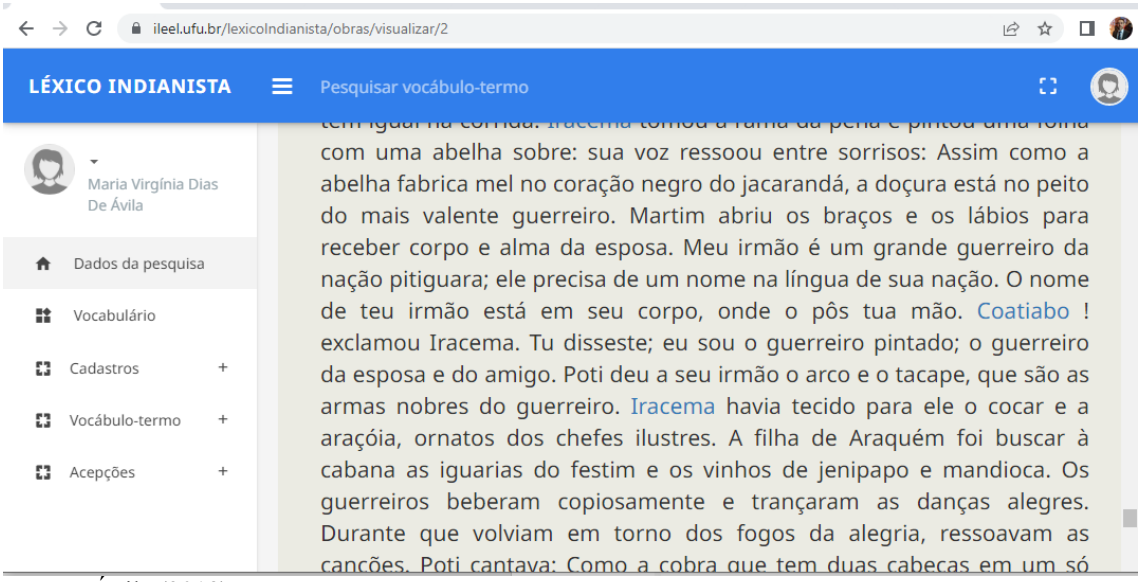
No modo leitura, ilustrado pela figura abaixo, temos os detalhes do verbete quando este é selecionado pelo usuário.

Figura 54 Interface leitura Léxico Indianista

Fonte: Ávila (2018)

Há nessa tela a acepção do verbete, sua classe gramatical e um breve emprego ou significado deste, bem como a frequência com que aparece no *corpus*. Ao lado existem algumas outras informações relacionadas às obras ou à obra na qual ele foi encontrado, como: o título da obra da qual ele foi retirado, nome do autor e um botão no formato de uma lupa que nos leva para o texto de origem do verbete. Abaixo mostramos uma ilustração da tela que é apresentada quando clicamos para visualizar a obra completa.

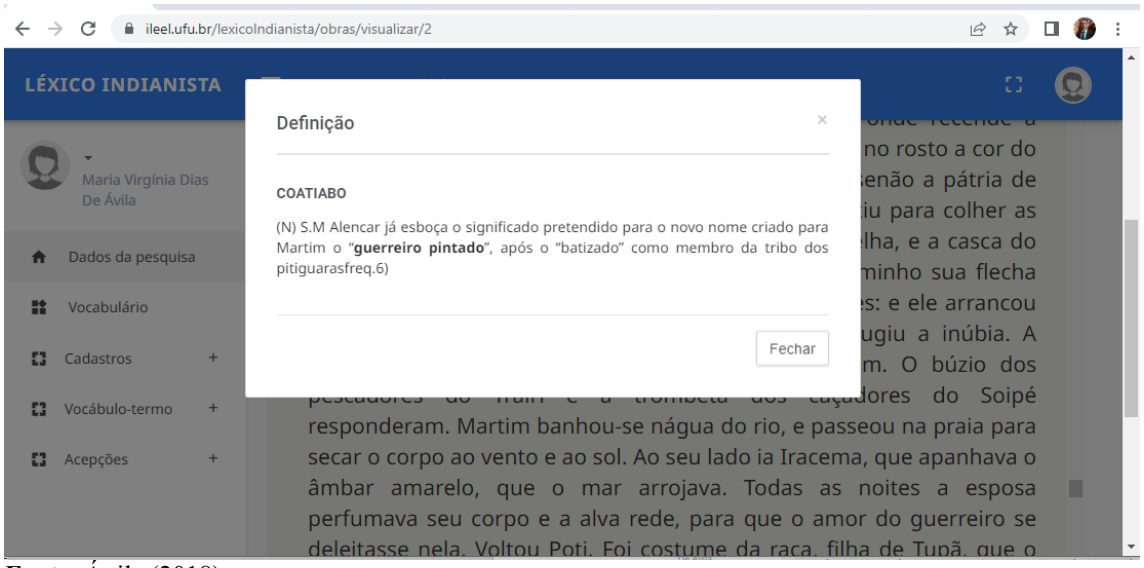
Figura 55 Modo leitura de obra



Fonte: Ávila (2018)

Na tela ilustrada pela figura anterior temos acesso ao texto da obra que escolhemos ler, ao lermos a obra estão destacados nela os verbetes que foram encontrados pela autora. Ao clicarmos, em um desses verbetes nos são apresentados em forma de um *pop-up* os detalhes desse verbete, com isso conseguimos trazer mais dinamismo para a leitura do texto, deixado disponível para o leitor o significado desse verbete e como ele foi empregado pelo autor no texto de origem, agregando assim mais informações para ele no momento da leitura.

Figura 56 Modo leitura com pop-up



Fonte: Ávila (2018)

Os dois módulos, tanto o Sertanista quanto o Indianista, dão para o leitor uma visão mais dinâmica na hora e lerem as obras, com a possibilidade de obter mais detalhes sobre os verbetes que lhe são apresentados, trazendo assim uma visão mais completa sobre a obra, mostrando para ele como esses termos foram empregados nos textos.

Falaremos a seguir do Léxico da Tabatinga, dando mais detalhes e ilustrações desse modulo.

4.3 LÉXICO DA TABATINGA

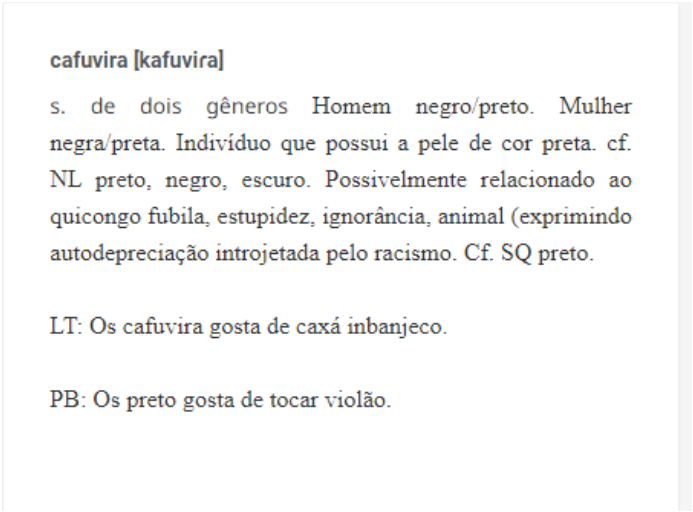
Descrevemos e demonstramos nesse capítulo as funcionalidades implementadas para o modulo Léxico da Tabatinga. Apresentamos a seguir a página inicial do Léxico. Diferentemente dos Léxicos Sertanista e Indianista, esse modulo não é composto por fazes, há apenas o modo vocabulário, com uma lista dos vocábulos-termos.

Figura 57 Home Léxico da Tabatinga

Fonte: https://www.ileel.ufu.br/lexicoTabatinga/

Quando escolhemos um verbete, e clicamos sobre ele na página inicial do módulo, temos algumas informações que são relacionadas a esse verbete e que nos são apresentadas, como podemos ver na **Figura 58**. Abaixo evidenciamos as informações gramaticais do verbete e o seu significado.

Figura 58 Informações gramaticais de verbete



cafuvira [kafuvira]

s. de dois gêneros Homem negro/preto. Mulher negra/preta. Indivíduo que possui a pele de cor preta. cf. NL preto, negro, escuro. Possivelmente relacionado ao quicongo fubila, estupidez, ignorância, animal (exprimindo autodepreciação introjetada pelo racismo. Cf. SQ preto.

LT: Os cafuvira gosta de caxá inbanjeco.

PB: Os preto gosta de tocar violão.

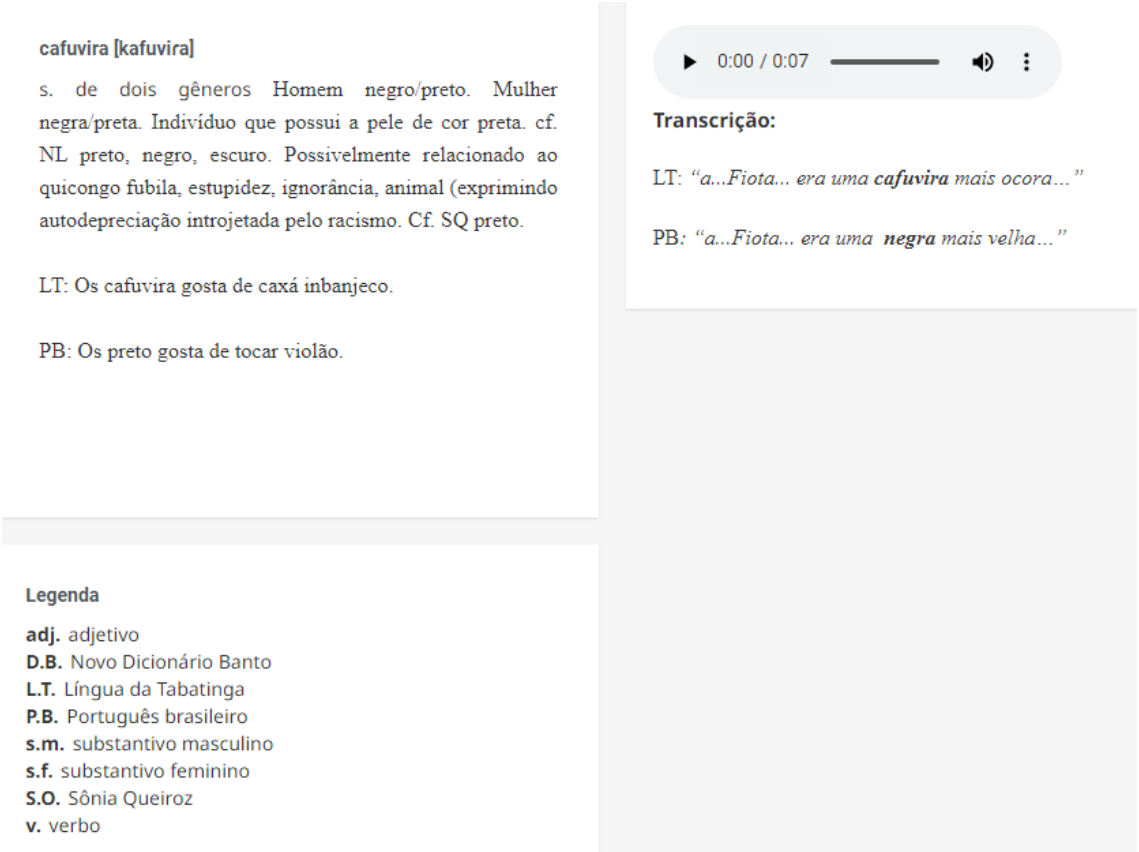
Fonte: https://www.ileel.ufu.br/lexicoTabatinga/

Temos também como informação complementar ao verbete um áudio gravado pela pesquisadora por meio de uma entrevista com alguns moradores da região e falantes da língua da Tabatinga juntamente com uma transcrição desse áudio.

Figura 59 Áudio do verbete

Fonte: https://www.ileel.ufu.br/lexicoTabatinga/

Figura 60 Detalhes do verbete



Fonte: https://www.ileel.ufu.br/lexicoTabatinga/

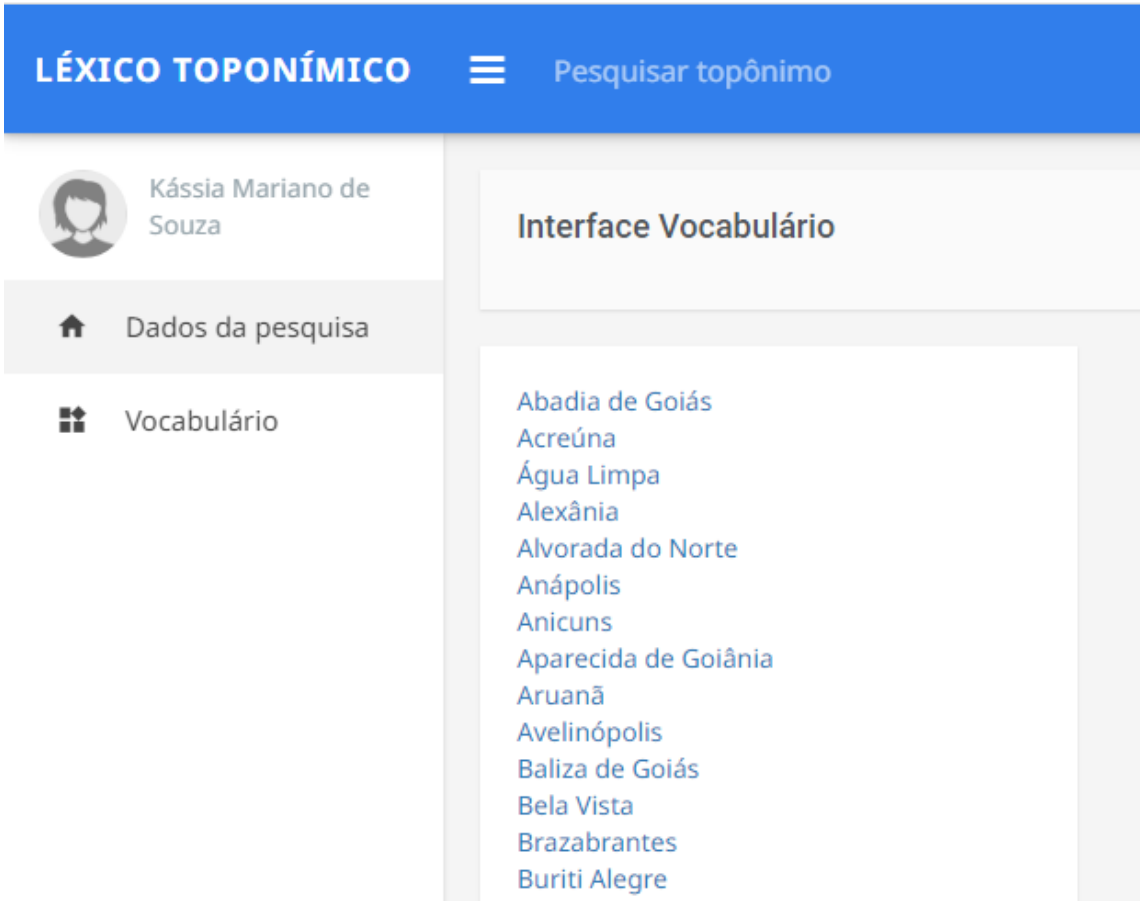
Explicamos a seguir sobre Léxico Toponímico de Goiás em Libras, dando mais detalhes e ilustrações desse modulo.

4.4 LÉXICO TOPONÍMICO DE GOIÁS EM LIBRAS

Esse módulo do sistema teve como objetivo criar um instrumento lexicográfico on-line com o registro dos sinais analisados pela pesquisadora, no intuito de promover a divulgação dos sinais registrados e de favorecer os processos comunicativos em Libras no Estado de Goiás, ou seja, como defendido pela autora: "Compreender como se realiza em Libras o léxico toponímico do Estado de Goiás por meio da apreensão, descrição, análise e registro dos topônimos goianos, com o suporte da Linguística de Corpus" (Mariano, 2022, p10).

Apresentamos, a seguir, a página inicial do Léxico Toponímico de Goiás em Libras. Diferentemente dos demais léxicos esse módulo não é composto por fases, há apenas o modo vocabulário com uma lista dos topônimos goianos.

Figura 61 Home Léxico Toponímico em Libras

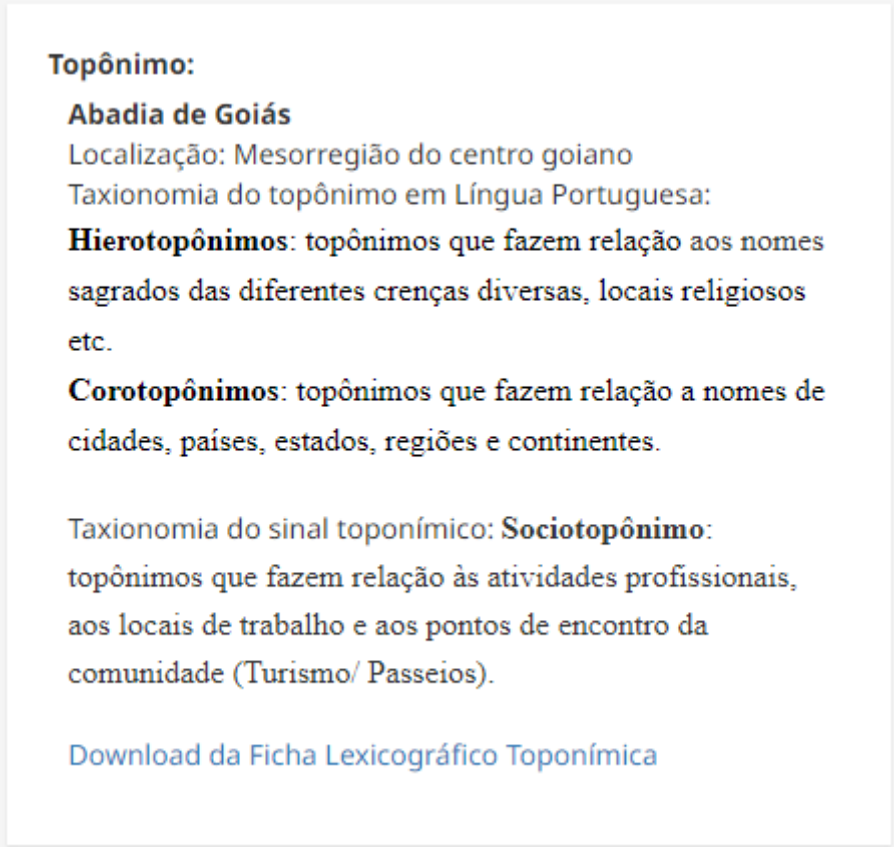


Fonte: https://www.ileel.ufu.br/topominiaLibras/home

Ao escolhermos um topônimo e clicarmos sobre ele na página inicial do módulo vocabulário, recebemos algumas informações que são relacionadas a ele e que nos são

apresentadas, como podemos ver na figura abaixo, em que há informações como: nome do topônimo, localização, Hierotopônimos, Corotopônimos e a Taxionomia do sinal toponímico, bem como um *link* para fazer o *download* da ficha lexicográfica do topônimo.

Figura 62 Descrição do Topônimo



Fonte: https://www.ileel.ufu.br/topominiaLibras/home

Também como informação complementar ao topônimo existe um vídeo gravado pela pesquisadora no qual ela traduz para a língua de sinais o topônimo selecionado e também são oferecidas algumas informações que são importantes na hora de realizar um sinal toponímico em libras como: descrição fonomorfológica do sinal, ponto de articulação, orientação da palma, movimento da mão, movimento dos dedos e expressões não manuais. Conforme pode ser visto na figura abaixo.

Figura 63 Descrição fonomorfológica do sinal

Fonte: https://www.ileel.ufu.br/topominiaLibras/home

Apresentamos abaixo o Léxico Machadiano, uma ferramenta para análise exploratória de *corpora,* dando mais detalhes e ilustrações desse modulo no capítulo que se segue.

4.5 LÉXICO MACHADIANO

Descrevemos nesse capítulo o módulo intitulado Léxico Machadiano, que tem como objetivo agrupar os *corpora* em um único local; o intuito desse módulo é dar ao leitor a possibilidade de realizar pesquisas lexicais em todos os textos que compõem a plataforma *GECon*.

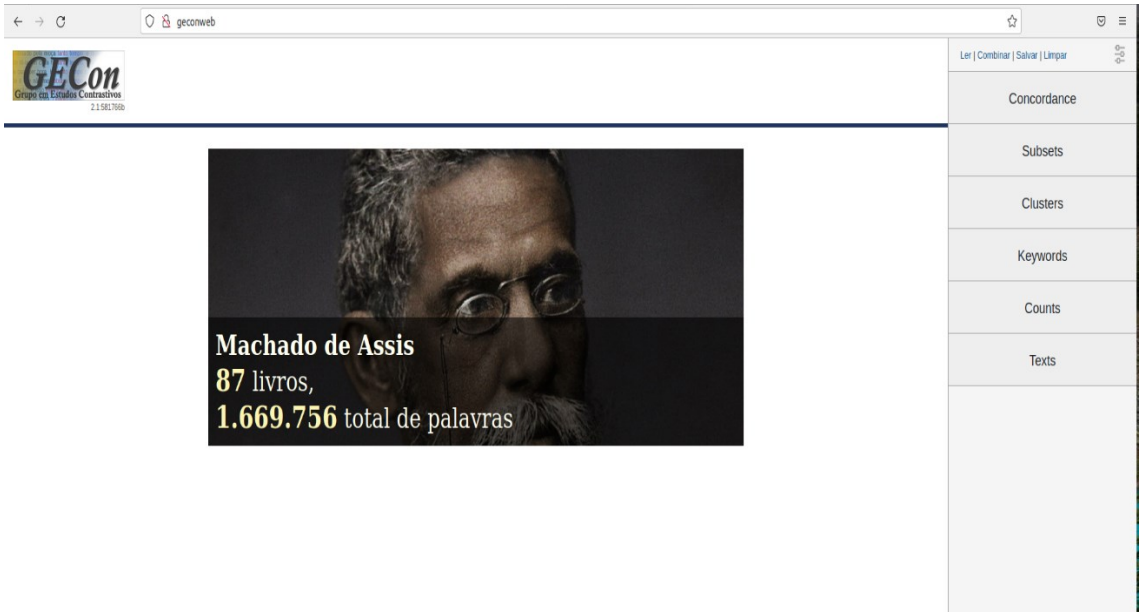
Por meio de funcionalidades como, *Concordance, Subsets, Clusters, Keywords e Counts* é possível realizar pesquisa exploratórias nos textos. Mostramos ao leitor como os métodos assistidos por computador podem ser usados para estudar textos literários, utilizando como base a linguística do *corpus*. A plataforma em questão pode possibilitar

ao leitor novos *insights*[54], tendo como base os resultados obtidos por meio das pesquisas realizadas nos textos disponibilizados nela.

O acesso ao Léxico Machadiano se deu por meio de um navegador *web*. Nesse caso, utilizamos o Google Chrome[55] como navegador. Como estávamos em um ambiente de desenvolvimento e de testes utilizamos o endereço que faz referência à máquina local de desenvolvimento.

Com o navegador de *internet* aberto, é só digitar o seguinte endereço http://localhost na barra de endereço para ter acesso ao sistema. Na tela inicial, há uma logo com a identificação do grupo de pesquisa na barra superior, abaixo existe um componente com efeito carrossel em que são mostrados os *corpora* que estão disponíveis na ferramenta com a quantidade de palavras e livros.

Figura 64 Página inicial do sistema



Fonte: O autor

Na página inicial do módulo há um carrossel[56], uma sequência de três *slides* dos *corpora* que foram carregados. No canto esquerdo, temos a logo do grupo de pesquisa, o que demostra o vínculo da pesquisa ao grupo *Gecon,* já no canto direito existe um menu

54 Traduzido do inglês: *Insight* é a compreensão de uma causa e efeito específicos dentro de um contexto particular. O termo pode ter vários significados relacionados: um pedaço de informação o ato ou resultado de compreender a natureza interna das coisas ou de ver intuitivamente uma introspecção. Fonte: https://en.wikipedia.org/wiki/Insight acessado em: 12. Dez 2022
55 Navegador de *internet* desenvolvido pela Google.
56 Carrossel é um *slide show*, uma sequência de *slides* com informações diversas.

de acesso rápido a algumas funcionalidades disponibilizadas pelo módulo, como: *Concordance, Subsets, Cluters, Keyword, Counts e Texts*.

Daremos alguns detalhes dessas funcionalidades a seguir.

4.5.1 Counts

Essa funcionalidade dá uma visão geral e interativa da obra selecionada ou dos *corpora,* com contagem de palavras nos *corpora*, livros e ou subconjuntos. Caso tenha selecionado um *corpus*, ela mostrará os textos que a compõem, a quantidade de parágrafos de cada texto, o total de palavras, de citações e de suspensões curtas.

Figura 65 Tela da ferramenta Counts

Showing 1 to 26 of 26 entries, 321,363 total de palavras, 2,189 nas citações , 319,153 sem aspa
suspensões longas

	Livro	Capítulos	Total palavras	In Quotes	In Non-quotes	In Short
1	A pianista	1	8,463	0	8,463	
2	Ao Acaso	1	77,620	0	77,620	
3	Cinco mulheres	5	5,538	0	5,538	
4	Comentários da semana	1	21,790	441	21,349	

Fonte: O autor

4.5.2 Concordance

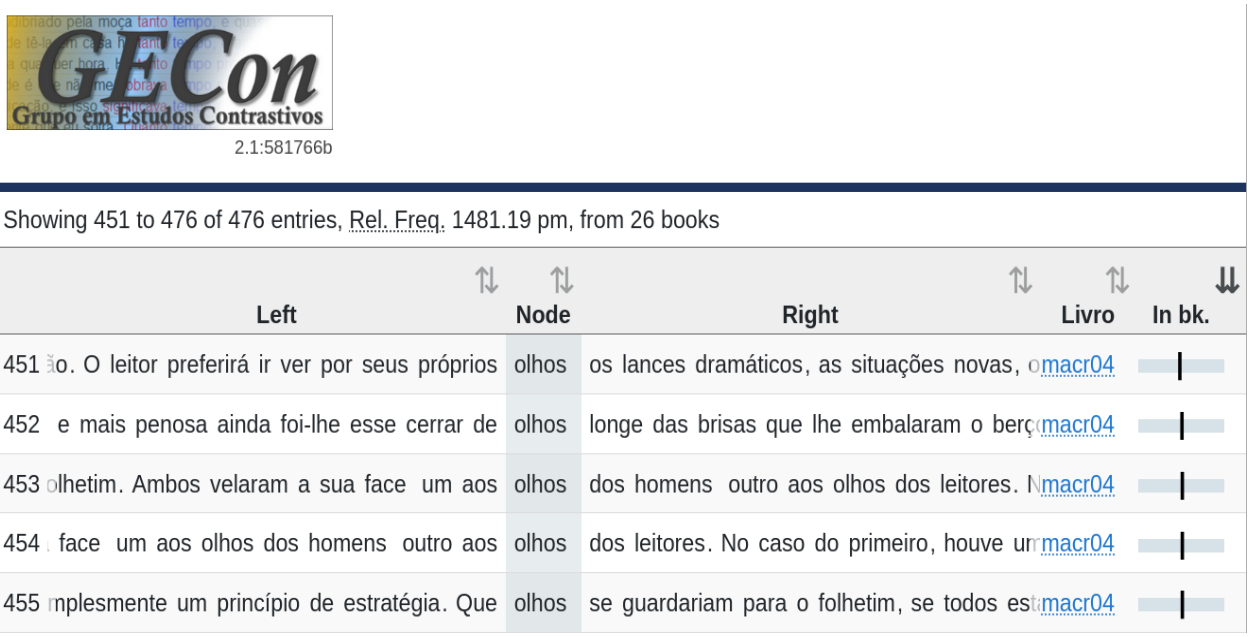
Clicando na guia Concordância o leitor levado à exibição de concordância. Para criar uma concordância ele precisará selecionar um *corpus* para pesquisar, a seleção é muito flexível e permite que ele possa escolher um ou vários *corpus*. Utilizamos esta opção quando queremos buscar no *corpus* correlações de concordância entre termos. Podemos realizar uma busca em todo o *corpus* ou em um texto em específico, se quisermos, podemos restringir mais a pesquisa e buscar por *Short Suspensions, Long Suspensions, Quotes e Non- quotes*[57]. Podemos também escolher como o resultado será

57 *Short Suspensions*: sentenças curtas, Long Suspensions: sentenças longas, Quotes: citações *e Non-quotes* sem citações

apresentado. Entre as opções, temos resultado simples, completo ou uma distribuição em *plot*, essa última mostra, de forma visual, a distribuição do termo pesquisado nos *corpora* ou no *corpus* em que foi pesquisado.

Buscar por termos, este é o parâmetro fundamental da pesquisa de concordância – que permite determinar a palavra ou a frase do nó que forma a base da concordância. A pesquisa só recuperará tokens válidos de acordo com as regras de associação, só buscará termos que coincidam com o informado no campo de pesquisa. Abaixo temos uma figura da tela exemplificando o resultado da pesquisa pelo termo "olhos" no concordânciador.

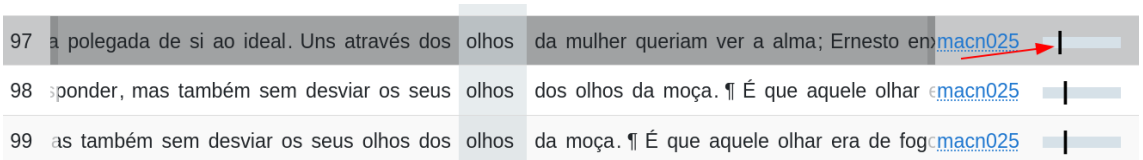
Figura 66 Tela do concordânciador



Fonte: O autor

A tela mostra o número máximo de palavras de cada lado em uma concordância, dependendo do comprimento das palavras e do tamanho da tela, o leitor pode ver menos, mas pode ver a visualização completa do capítulo clicando no botão *'In bk.'* (no livro) no final de qualquer linha.

Figura 67 Acessando o concordanciador no livro



Fonte: O autor

Ao clicar nesse botão o leitor é levado para o capítulo do texto onde o termo foi localizado, mostrando a visualização no texto do termo pesquisado.

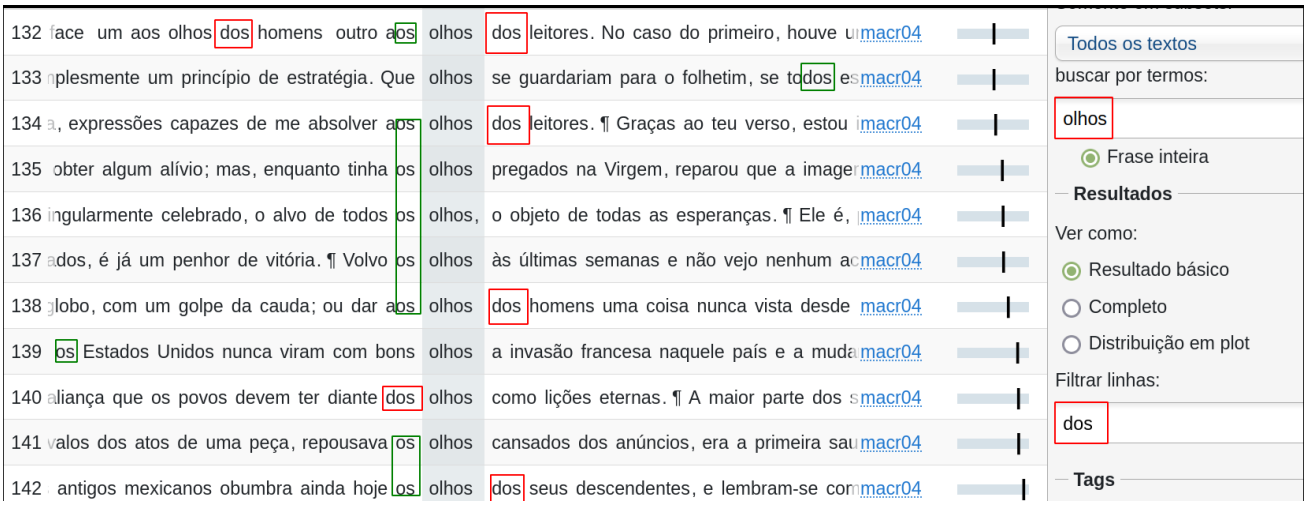
Figura 68 Ilustrando no texto o termo encontrado

Já por esta reflexão fica o leitor
instruído de que Ernesto não era homem de dar uma polegada de si ao ideal. Uns através dos **olhos** da mulher queriam ver a alma; Ernesto enxergou simplesmente uma bolsa recheada. Este modo de traficar a própria pessoa não é nenhuma descoberta, nem eu me dou por Arquimedes. Aponto simplesmente mais este traço do nosso herói.

Fonte: O autor

A opção f*iltrando linhas,* permite filtrar a saída do concordanciador pelas linhas que contêm uma sequência específica de letras (no nó e no texto). Por exemplo, procurar por olhos em Machado de Assis produzirá 451 resultados em 26 livros; quando usamos a opção 'filtrar linhas' e procuramos pelo termo “dos”, esse número é reduzido para alguns poucos resultados, conforme ilustrado a seguir.

Figura 69 Concordanciador filtrando por linhas



Fonte: O autor

Observemos que, ao pesquisarmos por uma sequência de caracteres, não necessariamente ele irá retornar todas as palavras completas, por exemplo: filtrar uma concordância de “dos” em Machado de Assis produzirá uma listagem como a ilustrada acima, podendo trazer a ocorrência do termo “dos” e “os” nas palavras.

A função de filtro é mais grosseira; pode ser aplicado de forma útil para filtrar um grande conjunto de resultados antes de fazer uma categorização mais refinada. Talvez o

leitor queira filtrar os resultados para linhas que contenham formas de palavras semelhantes, como por exemplo, filtrar por olho também recuperará linhas contendo olho e olhos. Além disso, ao contrário da pesquisa de concordância principal o filtro permite pesquisar tipos específicos de pontuação (por exemplo, aspas usadas em citações).

A visualização em *plot,* é um modo completamente distinto da demais, ela não mostra o texto em linhas de concordância, mas plota na tela uma distribuição das linhas de concordância referentes nos livros encontrados. Caso não seja encontrada nenhuma linha de concordância em um determinado *corpus,* ele não será mostrado no gráfico de distribuição em *plot*.

Figura 70 Gráfico de distribuição em plot

Fonte: O autor

4.5.3 Clusters

Sobre a utilização da funcionalidade de *clusters,* lembramos que *cluster* é o agrupamento dos dados quando não temos uma classe definida, ou seja, os dados com atributos similares são agrupados em categorias (Harrington, 2012, p 208). Quando queremos localizar nos *corpora* ou em um determinado *corpus* diferentes tipos de citações. Na pesquisa com os *clusters* são ignorados os limites delimitados pelo *subsets* e ela só prestará atenção na ordem do *token*[58]. Os *clusters* podem abranger citações mais longas, mas não considera capítulos inteiros.

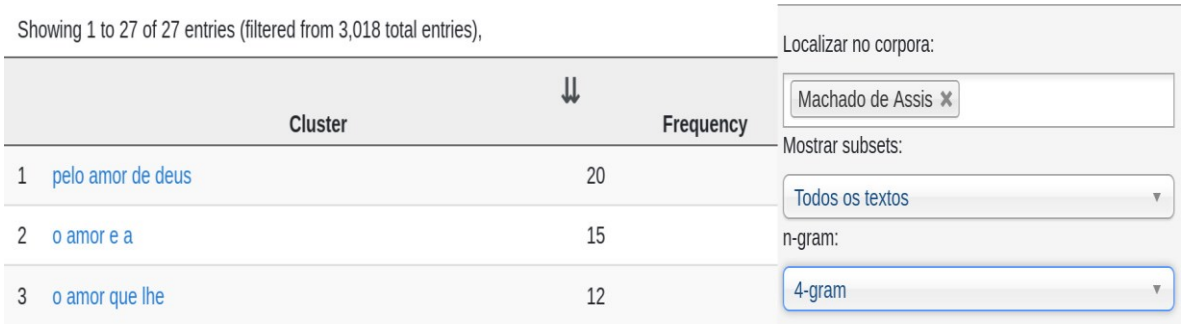
A saída da ferramenta *cluster* gera uma lista de frequência de palavras únicas 'clusters' (sequências repetidas de palavras). *Clusters* também são chamados de *grams*[59],

58 Token são identificadores, palavras-chave, operadores ou delimitadores lexicais.
59 Quantidade de termos ou *tokens* que são considerados como delimitadores da consulta.

onde 'n' representa o comprimento da frase. Se escolhermos um ‘1-gram’ (uma única palavra), recuperamos uma lista de palavras simples. Em Machado de Assis, por exemplo, as principais palavras recuperadas pela ferramenta foram, amor e Deus.

A seguir, mostramos uma figura ilustrando o resultado feito com a pesquisa por *clusters*. Nessa tela, há a classe e a frequência em quem se deu esse termo no *corpus*.

Figura 71 Tela da pesquisa por clusters



Fonte: O autor

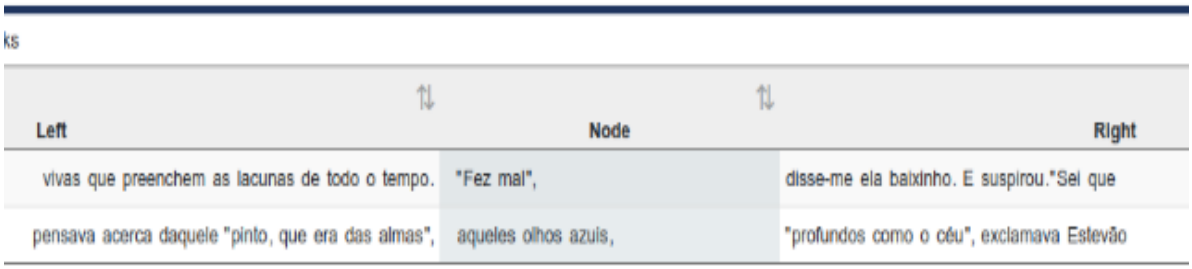
A ferramenta *clusters* permite restringir a pesquisa a um determinado subconjunto (‘Mostrar em *subsets*: selecione uma opção’) para que, por exemplo, leitor possa criar listas de frequência de um determinado subconjunto de dados.

4.5.4 Subsets

Utilizamos a funcionalidade de *subsets* quando queremos extrair dos corpora ou de um texto em específico do *corpus,* os subconjuntos que foram identificados pela ferramenta. *Subsets* são citações feitas nos textos, ou melhor, são palavras ou trechos dos textos que foram marcados por aspas.

A seguir, apresentamos, na **Figura 72**, a tela com a listagem dos *Subsets* identificados pela ferramenta. Por meio dessa ferramenta foi possível identificar, além das citações, os textos nos quais elas foram localizadas.

Figura 72 Tela do Subsets

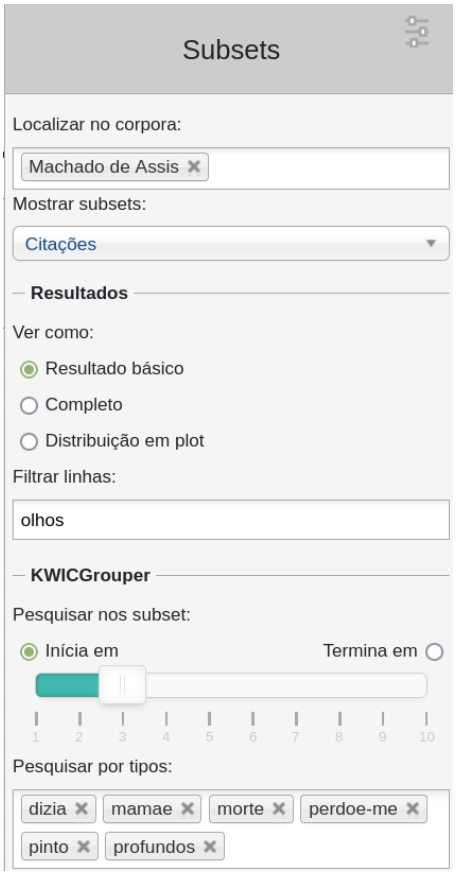


Left	Node	Right
vivas que preenchem as lacunas de todo o tempo.	"Fez mal",	disse-me ela baixinho. E suspirou."Sei que
pensava acerca daquele "pinto, que era das almas",	aqueles olhos azuis,	"profundos como o céu", exclamava Estevão

Fonte: O autor

Ao clicarmos no menu *subsets* da ferramenta temos acesso a alguns itens que podemos escolher para refinar a nossa busca como: escolher o *corpus* de busca, quais tipos de *subets* queremos que seja exibido, além de poder ver o resultado de três formas distintas.

Figura 73 Menu *subsets* da ferramenta



Fonte: O autor

Filtrando o resultado por linhas, a opção permite filtrar a saída pelas linhas que contêm uma determinada sequência de letras. O princípio de funcionamento dessa opção é próximo se não igual ao da ferramenta de concordância. Por exemplo, você pode filtrar

subsets por uma palavra em específicos como olhos, o resultado será uma lista com todos os textos em que o termo foi encontrado.

Alem da opção de filtro por linha podemos aplicar esse resultado retornado pela ferramenta a algumas outras restrições.

O *KWICGrouper* é uma ferramenta que permite agrupar rapidamente as linhas de um *subsets* ou do concordânciador de acordo com os padrões encontrados ao percorrer o item (Pesquisar por tipos), e ao escolher um dos itens apresentados nesses campos eles ficam em destaque nas linhas do *subset* onde ele foi encontrado.

Figura 74 Ferramenta KWICGrouper

Fonte: O autor

Apos a aplicação dos filtros por tipos um resultado, conforme o ilustrado abaixo, é retornado pela ferramenta.

Figura 75 Resultado após a aplicação do filtro KWICGrouper

e posição, — `homem,	dizia ela, que me viu a princípio com olhos avessos, pela diminuição que eu trazia à herança`.	No fim dizia que
volta? por que?` `Tua	mamãe disse ontem que papai está no céu`.	Helena levou as
nsava acerca daquele	`pinto, que era das almas`,	aqueles olhos az
`, aqueles olhos azuis,	`profundos como o céu`,	exclamava Estev
pel e leu estas linhas:	`Guiomar! Perdoe-me se lhe chamo assim; as convenções sociais condenam-me decerto,	

Fonte: O autor

A seguir, apresentaremos uma outra ferramenta chamada de *Keyword* e daremos mais detalhes no item abaixo.

4.5.5 Keyword

Essa é uma funcionalidade com um pouco mais de abrangência se comparada às outras duas, *subsets* e *cluster*. Com essa funcionalidade podemos realizar uma comparação por ocorrência dos termos entre dois *corpora* ou por dois textos. Ainda é possível tentar encontrar a ocorrência de um termo em dois diferentes *corpora* ou textos.

A ferramenta de *Keyword*[60] encontra palavras (e ou frases) que são usadas significativamente com mais frequência em um *corpus* em comparação com outro.

Figura 76 Tela da pesquisa por Keyword da ferramenta

Showing 1 to 50 of 553 entries,

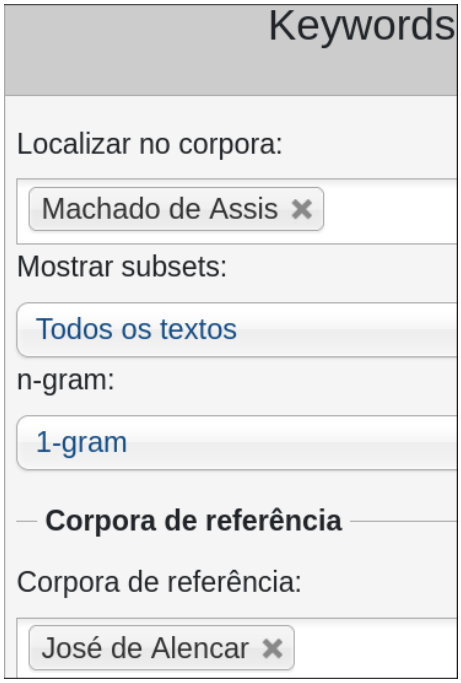
	N-gram	Target frequency	
1	helena	534	0
2	estacio	521	0
3	eu	1201	183
4	poeta	337	3
5	carta	339	5
6	nao	5059	1648
7	eduardo	224	0

Fonte: O autor

Além de poder comparar palavras únicas, a ferramenta também permite a comparação entre *clusters*. Considerando que a guia *cluster* se concentra apenas em um *corpus*, a função de *keyword* pode comparar listas de *cluster*. É possível selecionar dois *corpora* distintos, conforme pode ser mostrado na figura abaixo.

60 Keyword ou palavras-chave, tradução nossa.

Figura 77 Filtro de seleção de corpora da ferramenta *Keyword*



Fonte: O autor

Para essa ferramenta há a necessidade de selecionar dois *corpora* para a comparação, os *corpora* de estudo e o *corpora* de referência. O resultado da aplicação desse filtro pode ser visto na figura ilustrativa abaixo:

Figura 78 Resultado da pesquisa entro dois corpora

Showing 1 to 50 of 618 entries,

	N-gram	Target frequency	Ref frequency	LL	P
1	nao	30186	1648	457.34	p < 0.0001
2	ou	5756	142	362.41	p < 0.0001
3	eu	6402	183	344.93	p < 0.0001
4	ha	4595	126	259.45	p < 0.0001
5	nem	4533	129	245.49	p < 0.0001
6	rua	1061	1	172.15	p < 0.0001
7	mas	10955	604	160.5	p < 0.0001

Fonte: O autor

Ao compararmos utilizando como fator de comparação 1-gram entre o *corpus* de estudo Machado de Assis e o *corpus* de referência José de Alencar, temos o resultado exibido acima, ou seja, são contadas quantas vezes cada palavra aparecem nos dois

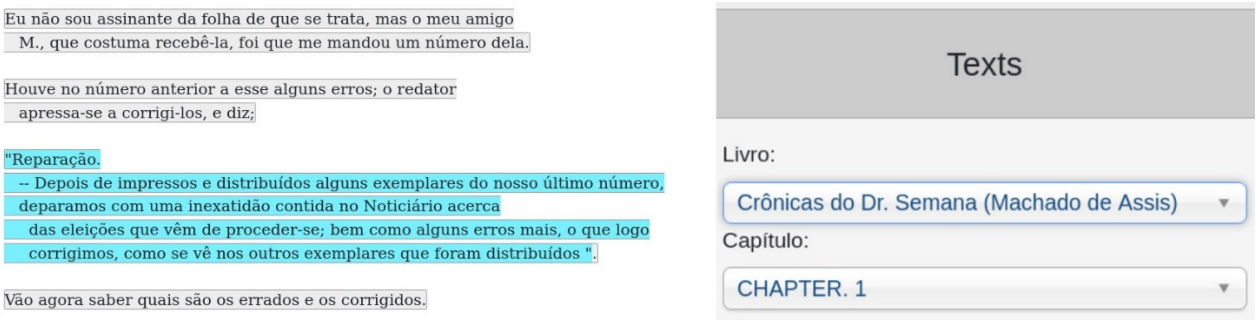
corpora. A saída da ferramenta *Keyword* é, por padrão, ordenada pelo valor de (LL[61]). Além disso, a ferramenta gera apenas as palavras positiva: aquelas que são mais utilizadas no *corpus* de destino do que no *corpus* de referência, por isso nem todas as palavras são contadas.

A seguir, falamos da ferramenta *texts* e, mostramos detalhes do seu funcionamento.

4.5.6 Texts

Essa tela nos permite ler a obra por completo, capítulo a capítulo e ainda há a opção de selecionarmos o capítulo que queremos ler e a de deixarmos em destaques as sentenças, as citações, as suspensões curtas, as suspensões longas e as citações embutidas. Cada uma dessas opções deixa o texto de cor deferente, destacando a opção escolhida.

Figura 79 Tela de leitura do texto



Fonte: O autor

A seguir, apresentamos as considerações finais do presente trabalho de dissertação, que busca resumir todos os caminhos percorridos até alcançar o presente momento, em que se faz necessário concluir os trabalhos, conferir os objetivos alcançados e considerar as dificuldades encontradas e os possíveis desdobramentos, em futuras pesquisas, dos aspectos aqui abordados.

61 *log-likelihood*, Máxima verossimilhança, é um método para estimar os parâmetros de um modelo estatístico. Assim, a partir de um conjunto de dados e dado um modelo estatístico, a estimativa por máxima verossimilhança estima valores para os diferentes parâmetros do modelo. Disponível em: https://pt.wikipedia.org/wiki/M%C3%A1xima_verossimilhan%C3%A7a Acessado em: 12 jan. 23.

CONSIDERAÇÕES FINAIS

Iniciamos o desenvolvimento desta pesquisa com o intuito de romper com uma barreira que muitos pesquisadores costumam encontrar após finalizarem a sua pesquisa. Depois de um árduo trabalho desenvolvendo a sua pesquisa, o pesquisador não consegue compartilhar com a comunidade os detalhes técnicos alcançados pelo seu trabalho. Seria interessante compartilhar com a comunidade, não apenas mostrando os resultados de seu trabalho, mas também dando a oportunidade para que eles possam testar na prática os resultados que foram obtidos com os dados, isto é, mostrando como os dados do seu trabalho podem ser usados de uma forma didática, prática e intuitiva. Dessa forma, esses resultados poderão servir de inspiração para outros pesquisadores.

A história vem nos mostrar que a evolução da LC está ligada diretamente à evolução da tecnologia, e está nos permite não apenas o armazenamento de *corpora*, mas também a sua exploração e manipulação. Atualmente, algumas atividades só são possíveis por causa da evolução tecnológica. As duas áreas, LC e Ciência da Computação, vêm crescendo e se desenvolvendo, uma servindo de insumo para a outra. Nesse sentido, vemos o quão importante é a utilização de ferramentas computacionais na LC e a LC na computação.

Revisitamos, a seguir, os objetivos de pesquisa, a fim de destacar os pontos que se configuram como relevantes para as conclusões e as considerações finais.

Como primeiro objetivo, propusemo-nos a testar as funcionalidades na plataforma durante seu desenvolvimento com os *corpora* compilados, no intuito de extrair dados que possibilitassem diferentes tipos de análises linguísticas. Acreditamos ter alcançado esse objetivo por intermédio da utilização do *corpus* do Machado de Assis como fonte de testes para tal finalidade.

O segundo objetivo foi disponibilizar para a comunidade acadêmica e interessados em geral a possibilidade de consulta dos dados que foram compilados por pesquisadores membros do grupo de pesquisa *GECon*. Esse propósito foi alcançado, com a criação da plataforma *GEConWeb*, que agrupa os Léxicos Indianista, Sertanista, Machadiano, Léxico da Tabatinga e o Léxico Toponímico de Goiás em Libras.

O terceiro e último objetivo foi possibilitar, por intermédio dessa plataforma, que as pesquisas desenvolvidas no âmbito do grupo de pesquisa *GECon* fossem amplamente divulgadas. Entendemos que alcançamos esse objetivo, uma vez que

disponibilizamos para acesso via *internet* o *site* do grupo de pesquisa, que pode ser acessado pelo seguinte endereço eletrônico: https://www.ileel.ufu.br/gecon/.

Nesse sentido, pensamos que os métodos, e os recursos que apresentamos nesta pesquisa possam ser de grande valia para aqueles que desejam conhecer um pouco mais dos trabalhos desenvolvidos no grupo de estudos *GECon*.

REFERÊNCIAS

ALENCAR, J. **Iracema**. Rio de Janeiro: Livraria José Olympio, Ed. Do centenário, 1965.

ALUÍSIO, S. M.; ALMEIDA, G. M. B**.** O que é e como se constrói um corpus? Lições aprendidas na compilação de vários corpora para pesquisa linguística. **Calidoscópio**, São Leopoldo, v. 4, n. 3, p. 156-178, set./dez. 2006. Disponível em: http://revistas.unisinos.br/index.php/calidoscopio/article/view/6002. Acesso em: 2 abr. 2019.

ÁVILA, Maria Virgínia Dias de**. Descrição etimológica do léxico indianista em José de Alencar:** uma análise lexicográfica direcionada por corpus. 2018. 255 f. Tese (Doutorado) - Curso de Letras, Instituto de Letras e Linguística, Universidade Federal de Uberlândia, Uberlândia, 2018. Cap. 4.

BARBOSA, J. L. N. et al. **Introdução ao Processamento de Linguagem Natural usando Python**, 1888.

BARBOSA, Maria Aparecida. Lexicologia, Lexicografia, Terminologia e Terminografia: identidade científica, objeto, métodos, campos de atuação. In: II SIMPÓSIO LATINO-AMERICANO DE TERMINOLOGIA. I ENCONTRO BRASILEIRO DE TERMINOLOGIA TECNO-CIENTÍFICA. **Anais...** Curitiba: IBICT, 1992.

BAKER, P. Corpus Methods in Linguistics. *In:* LITOSSELITI, L. (ed.). **Research methods in linguistics.** New York: Continnum International Publishing Group, 2010. p.93-113.

BAKER, P.; HARDIE, A.; MCENERY, T. **A glossary of corpus linguistics**. Edinburgh: Edinburgh University Press, 2006. https://doi.org/10.1515/9780748626908

BERBER SARDINHA, T. A influência do tamanho do corpus de referência na obtenção de palavras-chave. **DIRECT Papers 38**, São Paulo/Liverpool: LAEL PUCSP, p. 1-18, 1999. Disponível em: http://www2.lael.pucsp.br/direct/DirectPapers38.pdf. Acesso em: 22 jun. 2021.

BERBER SARDINHA, T. **Linguística de Corpus**. São Paulo: Manole, 2004. 410 p.

DUBOIS, J; GIACOMO, M.; GUESPIN, L.; MARCELLESI, C.; MARCELLESI, J.B; MEVEL, J.P. **Dicionário de linguística**. São Paulo, Cultrix, 653 p.

FERREIRA-BRITO, Lucinda. **Língua Brasileira de Sinais – LIBRAS** (Série Atualidades Pedagógicas). Brasil: Secretaria de Educação Especial, Brasília, 1998.

FERREIRA-BRITO, L. Por uma gramática de línguas de sinais. **Tempo Brasileiro**, UFRJ. Rio de Janeiro, 1995.

FROMM, G. O. Uso de Corpora na análise linguística. **Revista Factus**, São Paulo, v. 1, n. 1, p.69-76, 2003. Disponível em: http://www.ileel.ufu.br/guifromm/upload/ousodecorporanaproducaolinguistica.pdf. Acesso em: 19 jan.2022.

FIRESMITH, Donald G. et al. **Object-oriented requirements analysis and logical design:** a software engineering approach. New York: Wiley, 1993.

GALISSON, R; COSTE, D. 1983. **Dicionário de didáctica das línguas.** Coimbra, Livraria Almedina, p.763.

GÊ-ACAIABA, Roberta. **Processo de formação, inserção e permanência da língua da tabatinga na cidade de bom despacho:** investigação com suporte da linguística de corpus. Tese de doutorado (em fase de desenvolvimento), 2023.

HARRINGTON, P. **Machine Learning In Action**. Shelter Island, N.Y., Manning Publications Co., 2012

MARTINET, A. **Elementos de linguística geral**. 8. ed. Lisboa: Martins Fontes, 1978.

SHAW, Mary; GARLAN, David. **Software Architecture** - Perspectives on an Emerging Discipline. Prentice-Hall. 1996.

MANDEL, Theo. **Elements of user interface design**. New York: John Willey & Sons, 1997

MARIANO, Kássia. **Léxico Toponímico de Goiás em Libras.** Tese de doutorado (em fase de desenvolvimento), 2023.

McENERY, T.; WILSON, A. **Corpus linguistics**. Edinburgh, Edinburgh University Press. Edinburgh, Edinburgh University Press, 1996.

NELSON, M. Building a written corpus: What are the basics? *In:* O'KEEFFE, A.; MCCARTHY, M. J. (org.). **The Routledge handbook of corpus linguistics**. London: Routledge, 2010. p. 53-65. https://doi.org/10.4324/9780203856949-5

PIMENTA, Ana Paula Corrêa. **Representações do léxico sertanista em corpus da literatura regionalista brasileira:** protótipo de vocabulário etnoterminológico online. 219. 202 f. Tese (Doutorado) - Curso de Letras, Instituto de Letras e Linguística, Universidade Federal de Uberlândia, Uberlândia, 2019. Cap. 4.

PERRY, Dewayne E.; WOLF, Alexander L. Foundations for the study of software architecture. **Acm Sigsoft Software Engineering Notes**, [S.L.], v. 17, n. 4, p. 40-52, out. 1992. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/141874.141884.

PRESSMAN, R. S. **Engenharia de Software,** Rio de Janeiro: McGraw Hill, Tradução da 5ª edição, 2002.

PRESSMAN, ROGER S**. Engenharia de Software**. Mc Graw Hill, 6 ed, Porto Alegre, 2006.

PUSTEJOVSKY, J.; STUBBS, A. **Natural Language Annotation for Machine Learning**: Aguide to corpus-building for applications. Sebastopol: O'Reilly Media, 2012.

SHAW. M.; GARLAN, D. Software Architecture. **Perspectives on an Emerging Discipline**, Prentice Hall, 1996. In: SHAW, M.; GARLAN; D (Orgs) **An introduction to software architecture. Technical Report- CMU-CS-94166**, Carnegie Mellon University, January 1994.

SUMMERVILE, I. **Engenharia de Software**. 6ed. São Paulo: Editora Person Education, 2003.

SUMMERVILLE, Ian. **Engenharia de Software**. 9 ed. São Paulo: Pearson Education, 2011.

TAGNIN, S. E. O. **Corpora na Tradução**, São Paulo: Hub Editorial, 2015.

TAGNIN, S. E. O.; BEVILACQUA, C. **Corpora na Terminologia**. São Paulo: HUB Editorial, 2015.

WIDDOWSON, H.G. **Linguistics**. Oxford: Oxford University Press, 1996.

ZANETTIN, F. **Translation-driven corpora:** Corpus resources for descriptive and applied translation studies. London: Routledge, 2014. https://doi.org/10.4324/9781315759661